UNSUPERVISED VIDEO SEGMENTATION AND ITS APPLICATION TO ACTIVITY RECOGNITION

BY

HSIEN-TING (TIM) CHENG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Doctoral Committee:

Professor Emeritus Narendra Ahuja, Chair
Professor David Forsyth
Professor Mark Hasegawa-Johnson
Professor Emeritus Thomas Huang

ABSTRACT

We addressed the fundamental problem of computer vision: segmentation and recognition, in the space-time domain. With the knowledge that generic image segmentation introduces unstable regions due to illumination, compression, etc., we utilized temporal information to achieve consistent 3D video segmentation. By exploiting non-local structure in both spatial and temporal space, the instabilities of the segmented regions were alleviated. A segmentation tree was built within every frame, and the label consistency was enforced within each subtree (i.e. spatial clique). By roughly tracking 2D regions across each frame, temporal clique was built in which label consistency was enforced as well. The high-order (more than binary) Conditional Random Field (CRF) is designed and solved efficiently. Experimental results demonstrate high-quality segmentation quantitatively and qualitatively.

Taking segmented 3D regions, called tubes, as input, we developed an activity recognition framework not only to determine which activity existed in a video but also to locate where it happens. A robust tube feature was extracted with photometric and shape dynamics information. Activity was described as a Parts Activity Model (PAM) with a root template and four-part template under the root. Given the nature of the activity recognition problem that only some parts on the video were used to determine the activity label, we used Multiple Instance Learning (MIL) to formulate the problem. Latent variables included a tube index and the parts location under the root template. Experiments were conducted on three well-known datasets and a state-of-the-art result was achieved.

To my parents, for their love and support

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The holy grail of the intelligent vision system is that it is capable of detecting and recognizing objects and scenes, and further that it builds up knowledge within the process. One of the fundamental steps is image representation, that is, how it describes an image (or a set of images) such that it captures the photometric properties, geometric relation of objects of interest but also it is robust in handling all kinds of variation, e.g. scaling, lighting, motion. One dominant image representation is the interest point (patch) based approach [1]. It first searches for interest points which are more salient in terms of gradient and contrast, followed by describing the points with a local descriptor which captures local characteristics, such as SIFT [2] and HOG [3]. The Bag of Words (BoW) approach is then used to provide the statistics of local descriptors such as a histogram of word clusters [4, 5]. Here "word" stands for a certain representation of local descriptors. In other words, an image (or object of interest) becomes a fixed-length feature vector in terms of the number of words used. At the final stage, a trained classifier is applied to make the final decision as to whether it is detection or recognition.

Though the above mentioned pipeline demonstrates high-accuracy result over the years, we find local features may fail in the following aspects:
(1) Local feature are defined in terms of local gray-level variation, and thus, in general are susceptible to changes in imaging conditions (e.g. illumination and viewpoint).
(2) Recognition using local features helps find only the vicinity of the detected objects; it does not segment them.

Hence, a number of approaches use segmented image regions as image representation. Regions are rich descriptors of image properties, and unlike local features, allow for simultaneous object detection and segmentation.

Many obvious recognition criteria, such as those dealing with the extent and relative spatial layouts of object parts, are naturally captured by image regions in contrast to local features. Over the years, we have developed a line of works on region extraction and structural representation in terms of regions [6, 7]. We use gray-level intensity and contrast as the low-level properties to define and detect low-level image structures. We define an image region as a set of connected pixels surrounded by *ramp discontinuities*. We model ramp discontinuities with strictly increasing (or decreasing) intensity profiles. Each ramp discontinuity has a magnitude, or contrast, which allows us to associate a photometric scale with each boundary fragment surrounding regions as shown is Figure 1.1. We achieve a multiscale segmentation over a range of scales, by progressively removing boundary fragments whose photometric scales are less than the current scale of analysis. Finally, all regions detected at all photometric scales are organized into a tree data structure, segmentation tree, according to their recursive containment relations (illustrated in Figure 1.2).

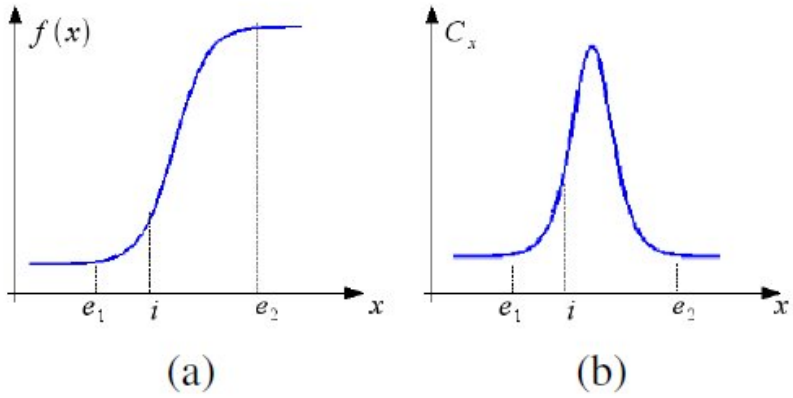


Figure 1.1: (a) Ramp model. (b) Ramp transform of $f(x)$. C_i is equal to $|f(i+a) - f(i-a)|$ where $a = \min\{|i - e_1|, |i - e_2|\}$ (from [7]).

Objects, in general, have hierarchical mutual relationships. Thus, an object category may be defined recursively in terms of object-part subcategories, which are objects in their own right. The recursive definition of an object includes the specification of photometric and geometric properties of parts and their spatial configurations. Hierarchical category definitions may also include the sharing of simple subcategories by more than one complex cat-

egory. For example, the category leg is shared by all legged animals, and, in turn, leg is an articulated combination of the simpler category of elongated shapes, which also occurs in the definitions of the categories of stools and scissors. It is reasonable to expect that simple categories (e.g., containing small/few/simple regions and structures) occur more frequently in real-world images, and their multiple occurrences exhibit smaller variations than encountered in more complex categories. Based on segmentation tree implementation and its extension, the above mentioned objects categorization concept can be realized [8].

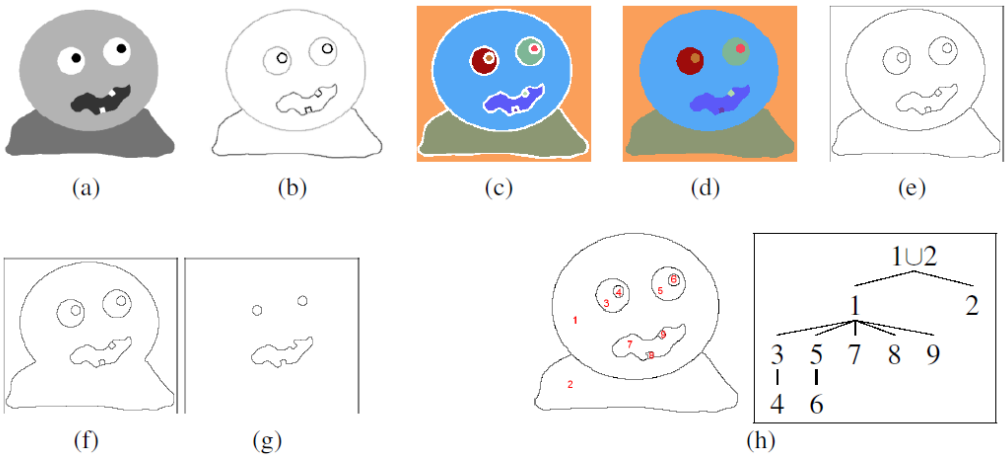


Figure 1.2: Illustration steps of a segmentation tree building algorithm. (a) Input image I. (b) Output of ramp transform, C, applied to I. Here, the darker the pixel, the higher the contrast of the underlying ramp. (c) Basins of C. Each basin is represented with a different color. These basins correspond to region seeds and the remaining pixels are ramp pixels (white color). (d) Final labeling obtained by growing the region seeds toward the ramp pixels using relaxation labeling. (e, f, g) Results of multiscale segmentation. (e) Segmentation result for photometric scale $= 5$. All regions are included. (f) Segmentation for $= 65$. Two regions (head and the body) merged. This means that the photometric scale of the boundary fragment in between the two merging regions is less than 65. (g) Segmentation for $= 80$. More regions have disappeared. The remaining regions are of a photometric scale larger than $= 80$, ensured by the region model and the algorithm. (h) Segmentation tree. On the left, each region is labeled by a number. Using the containment relations of regions, our algorithm computes the tree given on the right-hand side (from [7]).

Note that regions corresponding to low-contrast scene parts may some-

times merge or split under lighting or viewpoint changes, because the image acquisition process may capture or miss the associated low contrast boundary between them. This leads to segmentation instabilities across multiple images of the same scene. The key of achieving object recognition or further object categorization is to have a robust distant metric between two segmentation trees. It does not matter whether it is exact/inexact graph matching or tree matching, regardless of high computation complexity, the structure noise produced by unstable image segmentation prevents it from having a robust metric. Figure 1.3 is an illustration of segmentation instabilities. The first row is two images from a video sequence which are five frames apart; the second row shows their corresponding regions after segmentation; the third row zooms-in from the segmentation image focusing on the boundary between the man walking on the right-hand side and the horse. Clearly seen from the zoom-in parts, in (a3) there is a region boundary between the man and horse, however in (b3) no region boundary exists. If we build the tree from these two segmented regions, there will be two nodes (regions) with two different parents versus one node (at the zoom-in area).

With structure instabilities in mind, how can we utilize the richness of region representation to achieve recognition and segmentation at the same time? We move our target into video, a sequence of images with high space-time continuity. A fragment of region boundary may disappear in several frames for various reasons, but the fact that the region boundary coincides with the object boundary guarantees its existence at times in the sequence. It give us another critical dimension, the temporal dimension, to overcome instability of image segmentation. In Chapter 2, we will address the problem of video segmentation. By forcing non-local structural consistency, we can alleviate the drawback of region boundary instability and achieve an appealing segmentation result. Furthermore, we use segmented video units, called tubes, to model and recognize human activity.

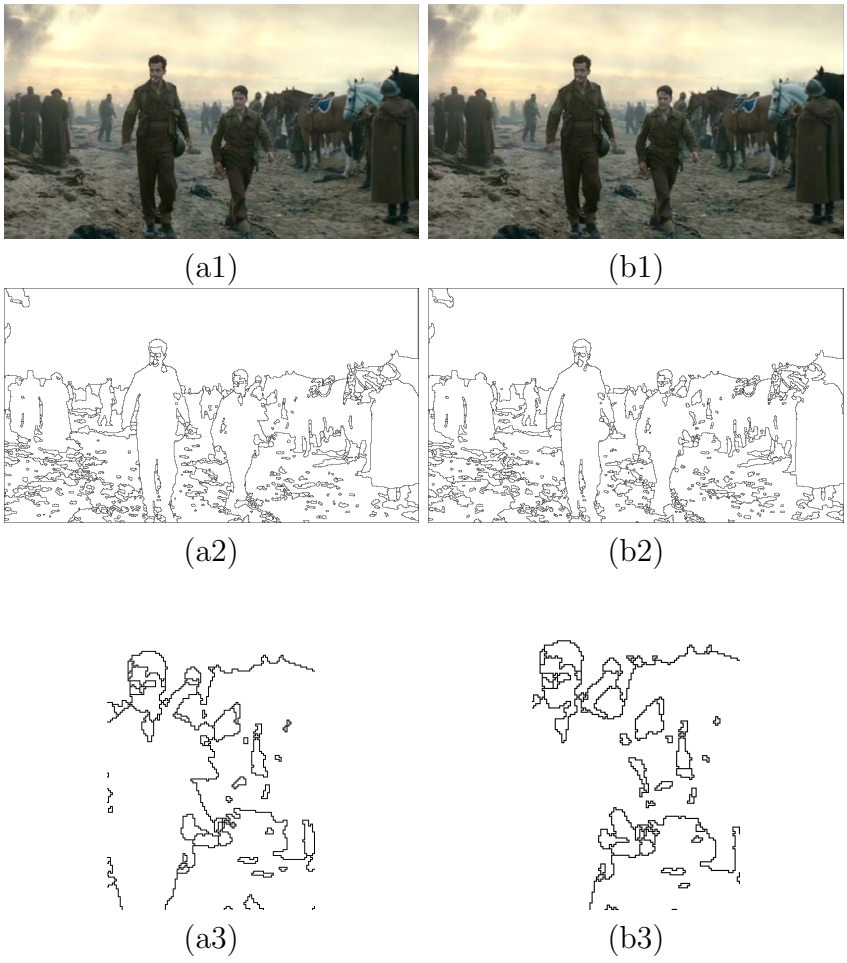


Figure 1.3: Segmentation instabilities.

CHAPTER 2

VIDEO SEGMENTATION

2.1 Introduction

Analogous to image segmentation, which partitions the image into groups of pixels with photometric similarity and outputs each group as a two-dimensional region, video segmentation partitions the three-dimensional spatiotemporal space into 3D regions (or region tubes), each having photometric coherence formed by the same region moving through consecutive frames. It is an important computer vision problem [9, 10, 11] with applications in areas such as activity recognition, video analysis, summarization, surveillance and browsing.

One line of research on video segmentation exploits grouping raw pixels across frames [12, 13, 14]. The pixels are represented as a multidimensional (space-time) feature, consisting of photometric, spatial and motion properties. However, this becomes infeasible, even for medium-sized videos. Thus, methods that use segmentation results of each image followed by tracking these regions have attracted much attention in recent years [15, 16]. These methods often produce less-desirable results due to the well-known lack of repeatability of image segmentation across frames. This decoupling of the video segmentation problem into two independent subproblems in space and time creates an unrealistic proposition. For example, regions in one frame across a low-contrast boundary may merge, and thus be undetected in the next frame. On the other hand, a large region that contains a slight variation of brightness may be split into a set of several smaller homogeneous regions in the subsequent frames. Some of the region tracking efforts tend to assume that (1) region properties are likely be the same in the consecutive frames, and (2) there exists a one-to-one correspondence for regions across

frames. In recent years, several methods have been proposed to deal with the above-mentioned problems [17, 18, 19]. Hedau and Ahuja [18] assumed that high contrast contours of an *object* are repeated along the video sequence, and then used contour-based partial matching to obtain many-to-many region correspondences. Bendel and Todorovic [17] extended the idea of partial contour matching in the video segmentation domain with an efficient matching process using the DTW (dynamic time warping) algorithm. The work of Grundmann et al. [19] treated video segmentation as a general clustering problem, in which a "region graph" from an over-segmented video volume is constructed, followed by hierarchical merging for generating coherent region boundaries.

Notwithstanding their demonstrated success, there is a major drawback in the tracking-based methods; namely, they must make hard decisions about identifying region correspondences and their merging/splitting for each pair of frames. Since we are concerned with bottom-up processing, lacking a model of the target being tracked, and therefore the nature of the best correspondences, the combinatorial nature of region merging/splitting makes finding optimal region correspondences between two frames an NP-hard problem. This implies that in a straightforward implementation, robust algorithms must take space-time constraints over many frames into consideration for segmenting regions and determining their correspondences. In the approach we present in this dissertation, we achieve such robustness while avoiding explicit and hard decisions made from local space and temporal information.

By formulating the video segmentation as a higher-order label consistency problem, we propose to solve the above problems via exploiting higher-order (instead of local) spatial and temporal structure. Following are the salient features and contributions of our approach:
(1) We treat each over-segmented region as a random variable. Random variables in different frames having the same label constitute a region tube. Hence photometric homogeneity within a region tube is achieved by enforcing *neighboring* regions with similar properties to take the same label.
(2) Instead of region adjacency, the neighbor definition in common use, we group each region with a larger set of higher-order neighbors, by forming its spatial and temporal *cliques*.

(3) We do not make hard decisions on region merging or splitting to form the spatial cliques. Similarly, our temporal cliques also include all pixel correspondences across several frames suggested by local motion.
(4) We allow multiple objects entering or leaving the scene, with no assumptions about the number of labels. Label creation and termination are determined by the data.
(5) We solve the label consistency and competition problem via a Conditional Random Field (CRF) formulation.

The work by Vazquez-Reina et al. [20] is one of the few algorithms that considers video segmentation as a labeling problem. They first enumerate multiple trajectories and treat each trajectory as a label, and then use CRF to solve the label consistency and competition problem. Our work bears some resemblance to theirs but differs in several aspects. First, they use multiple segmentation as well as superposition, while we apply multi-resolution segmentation and build a photometric tree as a spatial clique. Second, they prune the label space at the beginning by only allowing regions assigned to salient trajectories, while each region in our method can have its own unique label to make automatic label creation possible. Third, they assume corresponding regions in the consecutive frames must overlap with each other, while we use optical flow to locate the correspondences.

Overview of the Proposed Approach

Unlike the approaches that track regions across a pair of images, we simultaneously process a batch of frames to enforce spatial and temporal consistency. This is to reduce the accumulation of image segmentation errors that would be encountered in sequentially forming the region tubes. To formulate video segmentation as a labeling problem, we first construct a photometric segmentation tree for each frame by a multi-resolution segmentation algorithm [7]. The regions having the finest (lowest) contrast are considered to form the leaf level. Each leaf node is assigned a random variable X_i. Together the set of all regions across frames, and their labels, define a random field $\mathbf{X}$. The regions corresponding to those variables $\{X_i\}$ having a single label constitute a 3D region (tube).

One critical step of the proposed approach is the definition of the label space $\mathcal{L}$. While we make no assumption about the number of consistent photometric tubes present in a video, one possible extreme is that each leaf region contributes a unique label as an initial assignment. However, as described in Section 2.2.3, running time of the label inference process is quadratic in terms of label size $|\mathcal{L}|$. The above $\mathcal{L}$ definition is not feasible in practice. We proposed to use rough region correspondence obtained from optical flow along the frames to construct region tracks called *temporal cliques* as the label definition, i.e. each temporal clique contributes a label. Not all labels are available in each frame. A label would terminate if the corresponding region track vanished (e.g., object leaves the scene); a new label would be created if there is a region not belonging to any existing tracks (e.g., object enters the scene).

The consistent labeling problem is formulated as an inference process of a higher-order CRF. Conventional CRF formulation uses unary potential and binary potential to ensure label consistency. However, using these two potentials alone tends to *oversmooth* the labeling, resulting in unnecessary region merging. To overcome this problem without sacrificing the tractability of the inference process, we apply the higher-order potentials and the robust P^n model proposed by Kohli et al. [21] in the context of multi-class image segmentation. Each higher-order potential is defined on the set of regions forming a spatial and temporal "*clique*". A spatial clique represents a parental (non-leaf) node in the photometric tree, which is the union of several regions r_i in the leaf level. As indicated earlier, a temporal clique is formed by region correspondences across frames. The potential function that penalizes labeling inconsistency in each clique is determined by the photometric property and motion information. As shown in [21], a general submodular higher-order function can be transformed to a second-order function if the higher-order potential is defined using the P^n model. We apply the efficient graph-cut based α-expansion move algorithms of [22] to estimate the labels.

2.2 Higher-Order Consistent Labeling

In this section, we describe how to construct cliques and design the unary, binary and higher-order potential to achieve consistent labeling. We begin with the notation and definitions we use.

Preliminaries

Given a batch of frames $f_1, f_2, ...f_F$, a discrete random field $\mathbf{X}$ is defined over an index system $\mathcal{V}$ with a neighborhood system $\mathcal{N}$. Each random variable $X_i \in \mathbf{X}, i \in \mathcal{V}$ is associated with a leaf region in some frame. X_i would take a value from the label set $\mathcal{L} = \{l_1, l_2, ..., l_k\}$. Theoretically, we can set the $|\mathcal{L}| = |\mathcal{V}|$ so that each region can contribute a unique label l_i. In practice, we perform labeling pruning during the construction of a temporal clique to reduce redundant labels (detailed in Section 2.2.2). The neighborhood system $\mathcal{N}$ of the random field is defined by the sets $\mathcal{N}_i, \forall i \in \mathcal{V}$, where $\mathcal{N}_i$ denotes the sets of all neighbors of the variable X_i (where, for brevity, we loosely refer to those labels belonging to the neighboring regions as neighboring labels). Clique c is a set of random variables $\mathbf{X}_c$ which are conditionally dependent on each other. Both neighbors and cliques exist in two forms, spatial and temporal. Any possible assignment of labels to the random variables will be called a *labeling* (denoted by $\mathbf{x}$). The labels take values from the set $\mathbf{L} = \mathcal{L}^F$. A labeling $\mathbf{x}$ is interpreted as the estimated video segmentation. Leaf regions belonging to the same 3D segment are identified by the fact that the random variables associated with them take the same label. From [23], the posterior probablity $\Pr(\mathbf{x}|\mathbf{D})$ of CRF given observed data $\mathbf{D}$ is a Gibbs distribution and can be written in the form $\Pr(\mathbf{x}|\mathbf{D}) = \frac{1}{Z}\exp(-\sum_{c\in\mathcal{C}} \psi_c(\mathbf{x}_c))$, where Z is the usual normalizing constant known as the partition function, and $\mathcal{C}$ is the set of all cliques. The term $\psi_c(\mathbf{x}_c)$ is called the potential function of the clique c where $\mathbf{x}_c = \{x_i, i \in c\}$. The corresponding Gibbs energy is given by

$$E(\mathbf{x}) = -\log\Pr(\mathbf{x}|\mathbf{D}) - \log Z = \sum_{c\in\mathcal{C}} \psi_c(\mathbf{x}_c) \tag{2.1}$$

The MAP labeling $\mathbf{x}^*$ of the random field is defined as:

$$\mathbf{x}^* = \text{argmax}_{\mathbf{x}\in\mathbf{L}}\Pr(\mathbf{x}|\mathbf{D}) = \text{argmin}_{\mathbf{x}\in\mathbf{L}}E(\mathbf{x}) \tag{2.2}$$

Table 2.1: Variables annotations and descriptions.

variables	**descriptions**
A_S	spatial adjacency matrix (binary)
A_T	temporal adjacency matrix after flow ($f \rightarrow f+1$)
A'_T	temporal adjacency matrix after flow ($f \leftarrow f+1$)
$\mathcal{C}_T$	set of temporal cliques
c_t	temporal clique
R_{used}	regions have not included in c yet
r^f	region belongs to frame f
T_O	overlapping threshold

2.2.1 Spatial and Temporal Cliques

Conceptually, label consistency is achieved by penalizing variables having similar characteristics but taking different labels. In CRF we approach it by using penalty potentials. We propose to incorporate higher-order potentials to maintain label consistency:

$$E(\mathbf{x}) = \sum_i \psi_i(x_i) + \sum_{i,j \in \mathcal{N}_i} \psi_{i,j}(x_i, x_j) + \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c) \tag{2.3}$$

where c denotes a *clique* containing multiple regions. In conventional CRF the energy function consists of only the first two terms. If the neighborhood system $\mathcal{N}$ is defined according to region adjacency (both spatial and temporal), we have observed that a lack of higher-order terms tends to over-smooth the region tube and lose finer variation (as shown in Figure 2.1). Therefore ψ_c should be designed such that it allows but penalizes inconsistent labels within the clique. In this section we discuss how to construct the cliques with higher-order potentials, and then describe the formulation details of these potentials in the Section 2.2.2.

To achieve label consistency in both the spatial and temporal domain, we propose to construct two kinds of cliques: spatial clique c_s and temporal clique c_t. The descriptions of the annotations in the section are provided in Table 2.1. By using the multiple-resolution image segmentation algorithm [7]

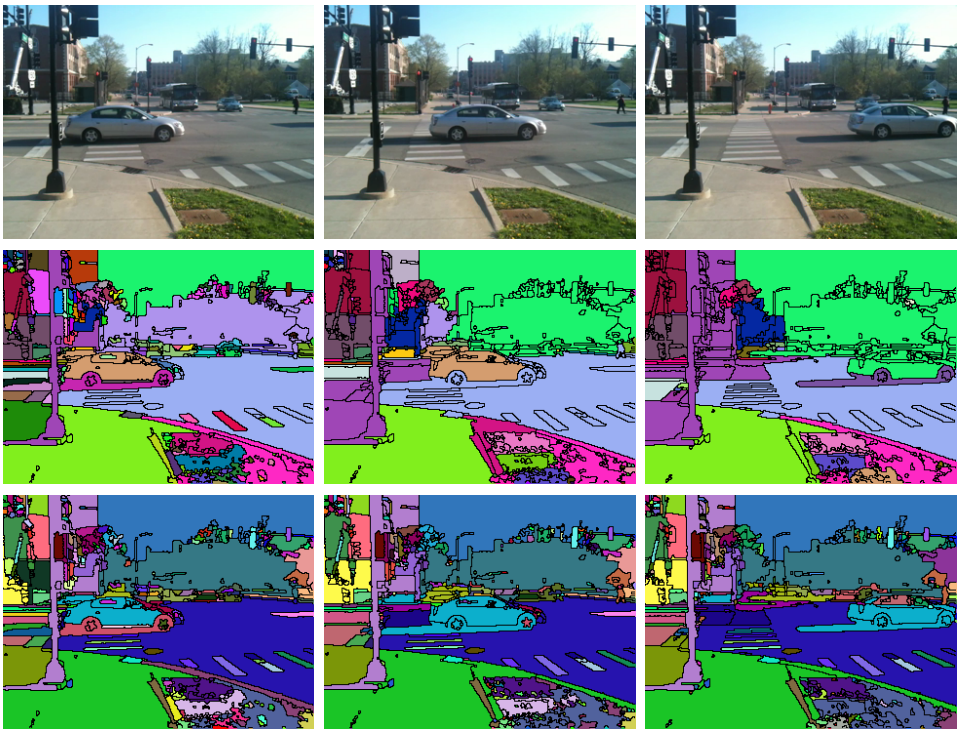


Figure 2.1: Comparison between binary potential and higher-order potential. First row: three sample frames from the original sequence. Second row: results with only unary and binary potentials. Third row: results using higher-order potentials. As can be seen in the second column, for binary potentials, the background building and tree are merged with the sky, and the zebra crossings are merged with the road. Even the car merges with the surroundings in the third column. By incorporating higher-order potentials, the salient details are retained (e.g., zebra crossing), while the low-contrast regions do merge (e.g., street light).

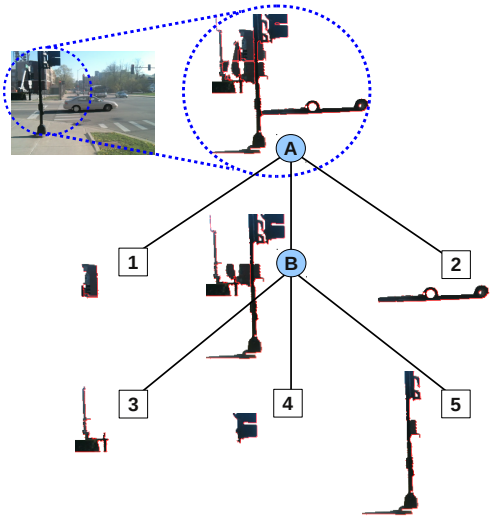


Figure 2.2: Spatial cliques and photometric tree. A and B are two interior regions in a photometric tree, each representing a spatial clique. Clique-A consists of leaf regions {1,2,3,4,5} and clique-B consists of {3,4,5}. It shows that a leaf region (four in this example) may belong to multiple spatial cliques.

to obtain the initial image segmentation, we are able to build a (photometric) tree in which interior regions are the union or merge of the children regions based on photometric similarity. As we traverse up the region hierarchy, the photometric variance within a region gets larger. Since the random field $\mathbf{x}$ is defined on regions at leaf level, for ancestor regions to form a single tube requires label consistency among its descendant leaf regions. Hence, each ancestor region in the photometric tree represents a spatial clique as illustrated in Figure 2.2. A leaf region r_i belongs to multiple spatial cliques c_s, each in general associated with a interior region.

As described in Section 2.2.1, the temporal cliques construction relates to label the space definition, and thus it is critical. Each temporal clique c_t can be viewed as a rough region track across frames guided by optical flow. The temporal adjacency matrix A_T, A'_T is first computed to estimate the region correspondence between a pair of frames,

$$A_T(i,j) = \frac{|r'_i \cap r_j|}{|r_i|} \tag{2.4}$$

Algorithm 1 $\mathcal{C}_T$ = *construct_Temporal_Cliques*

$\mathcal{C}_T = \emptyset$
for $f = 2, 3, ..., F-1$ **do**
    compute A_T, A'_T
    $R_{\text{used}} = \emptyset$
    for all $c_t \in \mathcal{C}_T$ **do**
        $R_{\text{used}} = R_{\text{used}} \cup \{r_i^{f-1} | r_i^{f-1} \in c_t\}$
        $c_t =$ *extend_Temporal_Clique*(c_t, A_T, A'_T)
    end for
    for all $r \in \{r^{f-1}\} \setminus R_{\text{used}}$ **do**
        $c_t =$ *create_Temporal_Clique*(r, A_T, A'_T)
        $R_{\text{used}} = R_{\text{used}} \cup c_t$
        C_T.add(c_t)
    end for
end for

Algorithm 2 c_t = *create_Temporal_Clique*(r, A_T, A'_T)

$c_{\text{prev}} = \emptyset$
$c_t = r$
while c_{prev}, c_t are not identitical **do**
    $c_{\text{prev}} = c_t$
    if for some r_j where $A_T(i,j) \geq T_O, r_i \in c_t$, & $A'_T(j,i') \geq T_O$ **then**
        $c_t = c_t \cup r_{i'}$
    end if
end while

Algorithm 3 c_t = *extend_Temporal_Clique*(c_t, A_T, A'_T)

for all $r_j \in$ next frame **do**
    if $A_T(i,j) \geq T_O$ or $A'_T(j,i) \geq T_O$ for some $r_i \in c_t$ **then**
        $c_t = c_t \cup r_j$
    end if
end for

where r'_i is the corresponding pixel of r_i after dense flow. Note that A_T is not symmetric, since it depends on the temporal direction i.e. forward or backward, thus A_T is for forward direction and A'_T is for backward direction. We use the dense flow algorithm proposed in [24]. The construction detail is provided from Algorithm 1 through Algorithm 3. Here we discuss the scenario of Algorithm 2 *create Temporal Clique*. Excluding the first frame, Algorithm 2 is called when there are regions where there is no correspondence; these unused regions R_{used} are then responsible for creating their own cliques. For

example, a car enters the scene from outside of the frame boundary, or a person turns from back to front and his/her face has not been seen before. As illustrated in Figure 2.3, a region may belong to multiple c_t due to noise in image segmentation or optical flow. In both spatial and temporal cliques, this one-too-many scenario leads to a competition between labels. A good design of the potential function would resolve the competition efficiently, assigning the best label to each region.

2.2.2 Potential Functions

Before defining the potential functions we first establish region-to-label relationships. The objective of potential functions is to penalize regions assigned an unlikely label, i.e. data term (*unary potential*), or neighboring regions are labeled inconsistently, i.e. smooth term (*binary and higher-order potential*). But at the same time, we prefer the discontinuity preserve result which respects the region boundary. As mentioned previously, a label is initiated as a specific temporal clique c_t. Hence, we build a model for each c_t and update it along the temporal clique construction process. The model we used here is a simple 2D histogram over the image space. A more complex model (e.g., appearance model) can be applied as well. Given a current histogram model h_t of a specific c_t, we estimate the likelihood function for a new joining region r as:

$$P_{h_t}(r) = e^{-\delta||hist(r)-h_t||} \tag{2.5}$$

where $hist(r)$ is a 2D histogram of r and $||hist(r) - h_t||$ is the max of $\mathcal{X}^2$ distance. Note the histogram $hist(r)$ is computed regarding optical flow motion. Then the model h_t is updated by:

$$h_t = \frac{h_t + P_{h_t}(r)hist(r)}{Z} \tag{2.6}$$

Z is the normalization term. By applying the above model update, after the temporal clique construction we have the clique model and the region-to-label likelihood function from Equation (2.5).

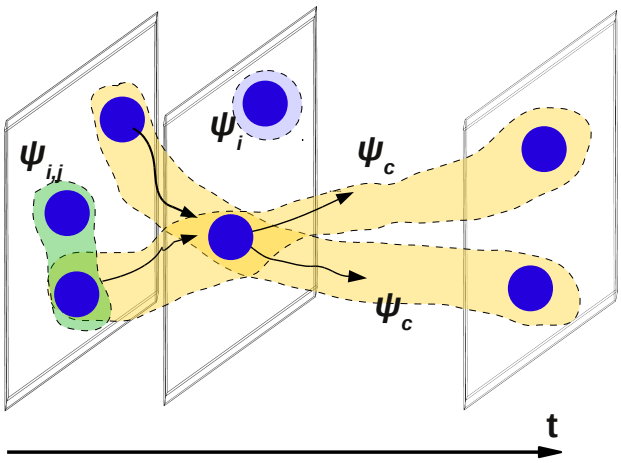


Figure 2.3: Potential functions and temporal cliques. Each leaf region is colored blue, and there are three kinds of potentials defined on it: unary ψ_i, binary ψ_{ij} and higher-order ψ_c. Here we show only the higher-order temporal potential, defined on a temporal clique (yellow). Each temporal clique is a result of inexact region tracking, therefore a region may belong to multiple temporal cliques.

Unary Potential

Though the label space may be large (depending on the size of temporal cliques $|\mathcal{C}_\mathcal{T}|$), actual choice for a region r is limited. Obviously the label set is at least the size of the temporal cliques it belongs to. Two other cases are:

• *case1 (temporal clique overlap)*: There exists a region $r' \in$ both c_t and c'_t where $r \notin c'_t$. Then the label corresponding to c'_t is incorporated to r's label set. It is illustrated in Figure 2.3.

• *case2 (under same spatial clique)*: If r and r' are under the same interior region, the corresponding labels of $\{c'_t\}$ where $r' \in \forall c'_t$ are added to r's label space.

Let x_i denote the label of r_i, and each label x_i has its corresponding temporal

clique c_{x_i} with model h_{x_i}. We define the unary potential as:

$$\psi_i(x_i) = \begin{cases} \theta_u \exp\{-\delta P_{h_{x_i}}(r_i)\} & \text{if } r_i \in c_{x_i} \\ \theta_u \exp\{-\delta \min(P_{x_i}(r_j), P_{x_i'}(r_j))\} & \text{if } r_i \in c_{x_i'}, r_i \notin c_{x_i}, r_j \in \text{both } c_{x_i} c_{x_i'} \\ \theta_u d_c(i,j) & \text{if } r_i, r_j \in \text{some } c_s, r_j \in c_{x_i} \\ \infty & \text{otherwise} \end{cases} \tag{2.7}$$

where $d_c(i,j)$ is the normalized ℓ_2-distance $\in [0,1]$ between r_i and r_j in the LUV colors pace, and θ_u is a constant.

Binary Potential

We use adjacency matrices A_S and A_T (and A_T') to refer to spatial and temporal region adjacencies. We assign the values $A_S(i,j) = 1$ if r_i and r_j are within the same frame and adjacent, and $A_{ij}^s = 0$ otherwise. A_T is defined as Equation (2.4). We define the following potential for each pair of spatially and temporally adjacent regions r_i, r_j:

$$\psi_{i,j}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ \theta_b \exp\{-\delta d_c(i,j))\} & \text{if } x_i \neq x_j \text{ and } A_S(i,j) = 1 \\ \theta_b \exp\{-\delta \max(A_T(i,j), A_T'(j,i))\} & \text{if } x_i \neq x_j \text{ and } A_T(i,j) \neq 0 \end{cases} \tag{2.8}$$

Higher-Order Potential

As mentioned earlier in Section 2.2, each interior region in the photometric tree determines a valid spatial clique c_s. However, not all such interior cliques (e.g., the root node which includes all leaf regions) should be expected to have high consistency among the labels of all the regions comprising them. As shown in Figure 2.2, although the car, the street light and the background building belong to one interior region, they should not all agree on the same label (e.g., due to their motion differences). The question then arises as to how to determine which subsets of labels are consistent. For example, how is a threshold set for the value of the variance within an interior region to de-

termine consistency. Likewise, criteria are needed to determine consistency among labels within the temporal cliques.

The above-mentioned problems with label consistency within cliques can be addressed by a potential function like the ones defined in the P^n model [21], where the model has been applied to the problem of supervised multi-class image segmentation. The P^n Potts model is defined in terms of clique c:

$$\psi_c(\mathbf{x}_c) = \begin{cases} \gamma_k & \text{if } x_i = l_k, \forall r \in c \\ \gamma_{\max} & \text{otherwise} \end{cases} \tag{2.9}$$

where $\gamma_{\max} > \gamma_k, \forall l_k \in \mathcal{L}$. Essentially it means that regions within a clique that could not have unanimous label agreement get penalty $\gamma_{\max}$, and get r_k if they all agree on label l_k. If γ_k are identical among different k, it enforces label consistency rigidly. For instance, if all but one of the regions in a clique take the same label, then the penalty incurred is the same as if they were all to take different labels. The robust P^n model is further proposed to relax the consistency constraint by letting the penalty grow upon a slope instead of a step function like in the P^n model. It is defined as follows:

$$\psi_c(\mathbf{x}_c) = \min\Big(\min_{l_k \in \mathcal{L}}(\mathbf{P} - f_k(\mathbf{x}_c))\theta_k + \gamma_k), \gamma_{\max}\Big) \tag{2.10}$$

where parameter $\mathbf{P}$ and functions $f_k(\mathbf{x}_c)$ are defined as following if it is a temporal clique:

$$\mathbf{P} = \sum_{r_i \in c} P_{h_k}(r_i), \forall k \in \mathcal{L} \tag{2.11}$$

$$f_k(\mathbf{x}_c) = \sum_{r_i \in c} P_{h_k}(r_i)\delta_k(x_i) \tag{2.12}$$

where

$$\delta_k(x_i) = \begin{cases} 1 & \text{if } x_i = k \\ 0 & \text{otherwise} \end{cases}$$

If it is a spatial clique, then:

$$\mathbf{P} = |c|, f_k(\mathbf{x}_c) = n_k(\mathbf{x}_c) \tag{2.13}$$

where $|c|$ is the number of regions in clique c, $n_k(\mathbf{x}_c)$ denotes the number of regions in c which take the label k in labeling $\mathbf{x}_c$, and $\gamma_k, \theta_k, \gamma_{\max}$ are potential function parameters which satisfy the constraints: $\theta_k = \frac{\gamma_{\max} - \gamma_k}{Q_k}$ and $\gamma_k \leq \gamma_{\max}, \forall l_k \in \mathcal{L}$. Here we design γ_k to be inversely proportional to the "inseparability" between the corresponding region r_k and the clique c: $\gamma_k = \exp(-\mathcal{I}_c(r_k))$. $\mathcal{I}(\cdot)$ is the measurement of inseparability and defined as the average contrast along the boundary. In a spatial clique, it is the contrast along the region boundary. In a temporal clique, it is the contrast between the overlapping pixels in the consecutive frames; the less the contrast, the harder it is to separate the regions. The truncation parameter Q_k controls the rigidity of the potential and here we set $Q_k = \frac{|c|}{2}$.

2.2.3 Inference

Once we have defined unary, binary and higher-order potentials for the objective function in Equation (2.3), the optimal labeling $\mathbf{x}^*$ is obtained by the CRF energy minimization inference process Equation (2.2). For this we use the algorithms presented in [21] which show that the *move energy* function of higher-order potentials in the robust P^n model can be transformed to submodular quadratic, hence the energy minimization is achieved by a series of α-expansion moves which can be solved efficiently by st-mincut algorithm [22]. The total inference time is: num(cycles)$\times$num(iterations)$\times T$(st-mincut), i.e. $O(|\mathcal{V}|^2|\mathcal{L}|^2\log(\frac{|\mathcal{V}|^2}{|\mathcal{L}|}))$ in the worst case where $|\mathcal{V}|$ is the number of variables and $|\mathcal{L}|$ is the number of labels. In practice, however, num(cycles) can be considered as a constant and dropped from $O(|\mathcal{V}|)$ yielding $O(|\mathcal{V}||\mathcal{L}|^2\log(\frac{|\mathcal{V}|^2}{|\mathcal{L}|}))$.

2.3 Implementation Details

To speedup the segmentation process, there are two good options to reduce the leaf region count $|\mathcal{V}|$ or label size $|\mathcal{L}|$. For the first, we divide the video into overlapping chunks in which segmentation can be performed in parallel followed by merging of the result. The overlapping frames between chunks are used to propagate the labeling results across chunks. The resulting loss in label optimality may not be significant since mutual influence among frames decreases with their distance. We use chunks of 10$\sim$15 frames with one frame

overlap. For the second option, given the segmentation within each chunk, we observe that the CRF inference process can be further decomposed into multiple inference sub-processes, as long as their label spaces are disjoint. For example, a temporal clique c_i at the upper-left corner is disjoint with another temporal clique c_j at the lower-right corner, i.e. $c_i \wedge c_j = \emptyset$. Therefore the original label space $\mathcal{L}$ is partitioned into disjoint label subspace $\{\mathcal{L}_i\}$, the inference is performed in each $\mathcal{L}_i$ and the results combined afterward. With these speedup operations, it takes 5 seconds to segment each frame on average (not including the optical flow computation).

Table 2.2: The precision (PR) and recall (RE) rate of foreground and background on the Weizmann activities [25] image sequence. The first row is the activity category. Comparison is made between the mean shift segmentation algorithm and the proposed method without using spatial higher-order potentials. Note that each activity has 10 different subjects. The reported rate is the average over all subjects.

	bend				jack				jump			
	Foreground		Background		Foreground		Background		Foreground		Background	
Methods	PR	RE	PR	RE	PR	RE	PR	RE	PR	RE	PR	RE
Mean shift (%)	62	42	71	55	63	65	66	67	56	41	63	77
CRF - SP (%)	82	73	83	85	75	80	83	84	87	67	85	90
CRF (%)	**97**	**83**	**96**	**96**	**89**	**94**	**93**	**93**	**99**	**80**	**95**	**96**

pjump				run				side				skip			
73	68	70	65	61	50	67	71	71	62	72	62	75	50	70	72
89	80	85	78	90	78	88	80	80	76	77	70	87	70	87	80
95	**93**	**94**	**94**	**99**	**80**	**92**	**93**	**94**	**90**	**93**	**93**	**99**	**75**	**95**	**96**

walk				wave1				wave2			
65	53	71	65	61	58	68	74	64	67	71	68
86	69	80	78	81	77	80	75	85	77	82	81
99	**81**	**92**	**93**	**94**	**88**	**91**	**91**	**99**	**87**	**92**	**92**

2.4 Experimental Results

We compare our approach (abbreviated as CRF in the context) against three other approaches: mean shift, state-of-the-art Grundmann's graph-based grouping approach [19] and our approach without using higher-order spatial potentials (abbreviated as CRF $-$ SP). The potential parameters are set the same way across all video sequences: $\theta_u = 1000, \theta_b = 50, w_i = \hat{w}_i = 0.5$ for $i = 1, 2$. They are determined by a preliminary human sanity test. Since no benchmark for generic video segmentation is available, we conduct

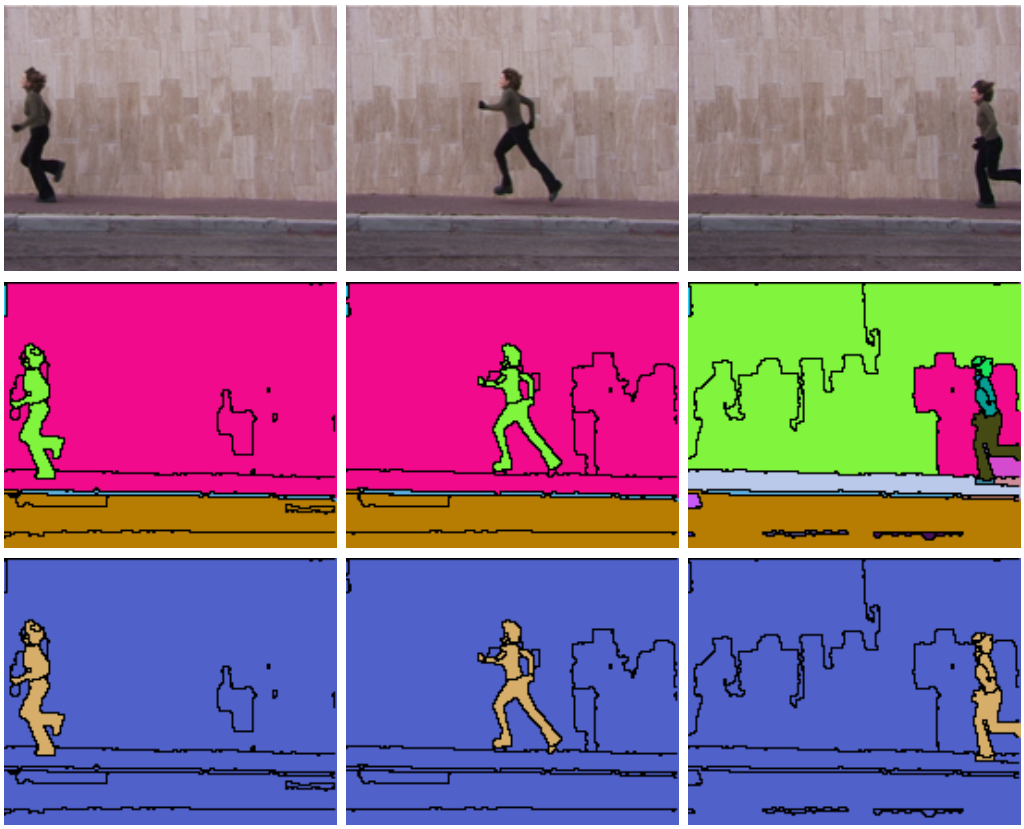


Figure 2.4: Video segmentation results for the *run* sequence from the Weizmann activity dataset. Top row: three consecutive frames (right to left) from *lena run2* activity; middle row: results using the mean shift segmentation algorithm; bottom row: results of the proposed method. The boundaries of the leaf regions are shown to demonstrate that the proposed method can overcome the instability arising from image segmentation. The background is separated into several, non-repetitive, irregular shape segments, and so is the foreground. Although the mean shift algorithm segments the left two frames successfully, it mislabels the left part of the wall as foreground in the right frame.

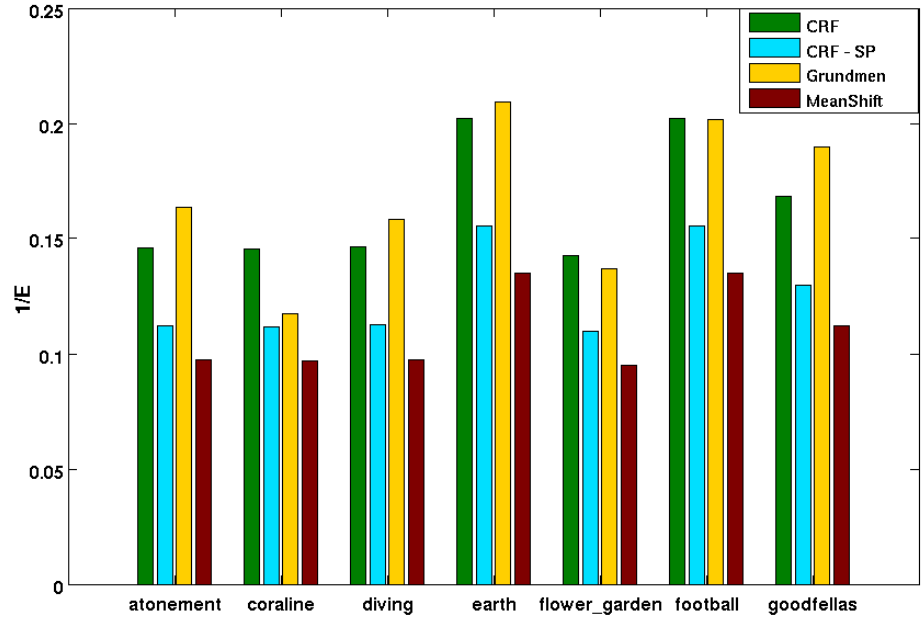
our experiments with standard datasets to evaluate the work both quantitatively and qualitatively. First we report the precision and recall on the Weizmann activity [25] dataset in Table 2.2. It consists of 90 videos: 10 distinct human activities (e.g., bend, jump, run), each with nine human subjects with foreground and background ground truth labeling. The average precision and recall rates over all videos are shown in Table 2.2. The rates are computed in terms of the total image area across all frames correctly segmented as foreground and background. For a given region tube with the same label (or the regions within the same cluster for the "Mean Shift" approach algorithm), we classify it as foreground if the majority of the covered area is foreground, and vice versa. From Table 2.2, it is clear that the CRF method outperforms the mean shift by a significant margin. It also demonstrates the role of higher-order spatial structure in achieving label consistency. Examples of the foreground and background segmentations obtained by our algorithm are shown in Figure 2.4. We show the boundaries of each leaf region to show how regions are merged by our algorithm. For the following more textured image sequences, we do not display the boundary to make the figure less noisy.

We next evaluate the methods on 15 more complicated videos as depicted in Figure 2.5. Since humans tend to perform object-level segmentation, producing ground truth for the textured image sequence via human annotation may not fully capture the performance. Performance evaluation on generic image/video segmentation has been discussed in [26, 27]. In this experiment, we use entropy as a measure $\frac{1}{E}$, computed as [28]:

$$E = H_r(I) + H_l(I) \tag{2.14}$$

where $H_r(I) = \sum_{j=1}^{N}(\frac{V_j}{V_I})H(R_j)$ denotes the expected *intra region tube entropy* as the sum of individual region tube entropies (weighted by volume), and $H_l(I) = -\sum_{j=1}^{N}\frac{V_j}{V_I}\log\frac{V_j}{V_I}$ denotes the *layout entropy* which is used to penalize over-segmentation. Higher $\frac{1}{E}$ value indicates better quality. Since [19] provides a multi-resolution (coarse to fine) segmentation result, we select the layer with the most similar number of 3D regions (i.e. labels) to compare with. The values of $\frac{1}{E}$ for each method for the 15 videos are shown in Figure 2.6. Our CRF method and [19] consistently outperform mean shift and

Figure 2.5: Example frames from video sequences. Top to bottom, left to right, the original sequences are: {atonement, coraline, diving, earth, flower garden, football, goodfellas, sufer, foroldmen, publicenemies1, publicenemies2, slumdog, UI traffic, waterski, ice skate and run example from Wiezeman dataset}.



(a) a

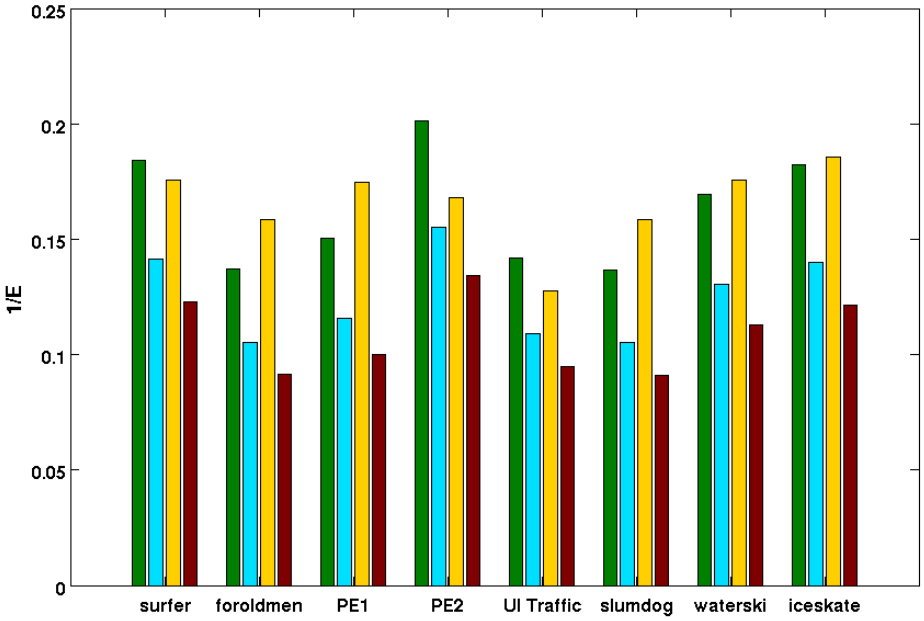


Figure 2.6: Quatitative measurement $1/E$ among four methods (CRF, CRF−SP, Grundmann [19] and Mean Shift) across 15 videos depicted in Figure 2.5. Higher value indicated better quality.

CRF−SP. While competing head-to-head against [19], CRF does better in 6 out of 15. Qualitative segmentation results of *waterski, goodfellas, iceskate, nocountryman* are shown in Figure 2.7. In the *ice skate* sequence, all the major parts (i.e., legs, body, hand, ground and back advertisement panel) are consistently segmented across frames. Note that from the first to the second frame, a new label for the text "OLYMPUS" is created. While in the *foroldmen* sequence, not only is the person consistently segmented, the new label corresponding to car explosion is being created. For the classic garden sequence shown in Figure 2.8, we also add comparisons with [17]. Note that these are three different approaches: region tracking [17], graph-based grouping [19] and our own method which is based on labeling many frames simultaneously using a conditional random field representation. Our algorithm (column (b)) erroneously segments part of the garden in the bottom frame as tree. However, for the remaining parts, our method outperforms the other two methods; [17] does not track any of the regions except the tree, and [19] simply over-merges the regions ("tree and hous" is one regions, the "grass area and flower" is another). More importantly, the shift between corresponding regions (indicated by the movement of colored regions) in our method is highly correlated with the image motion.

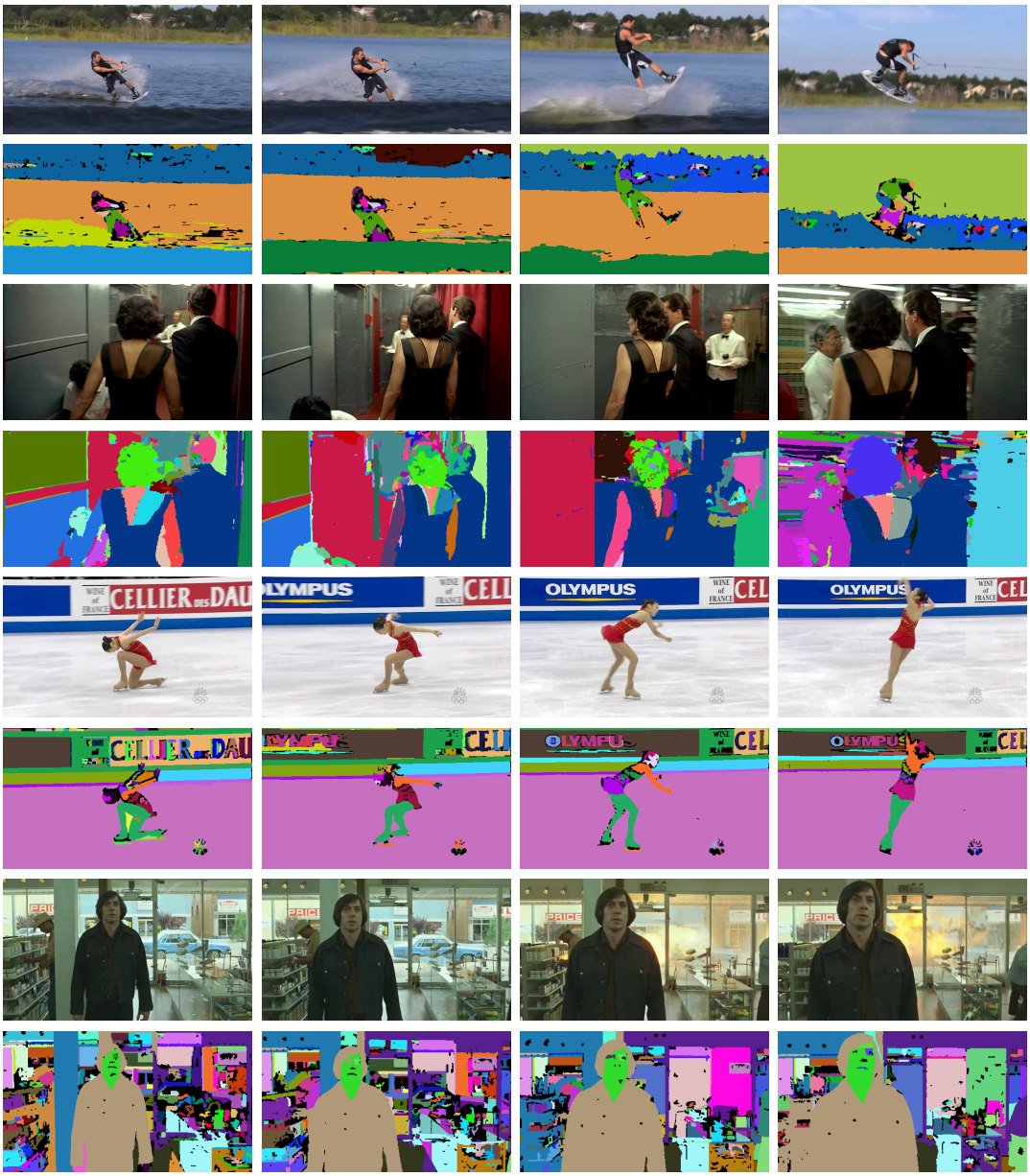
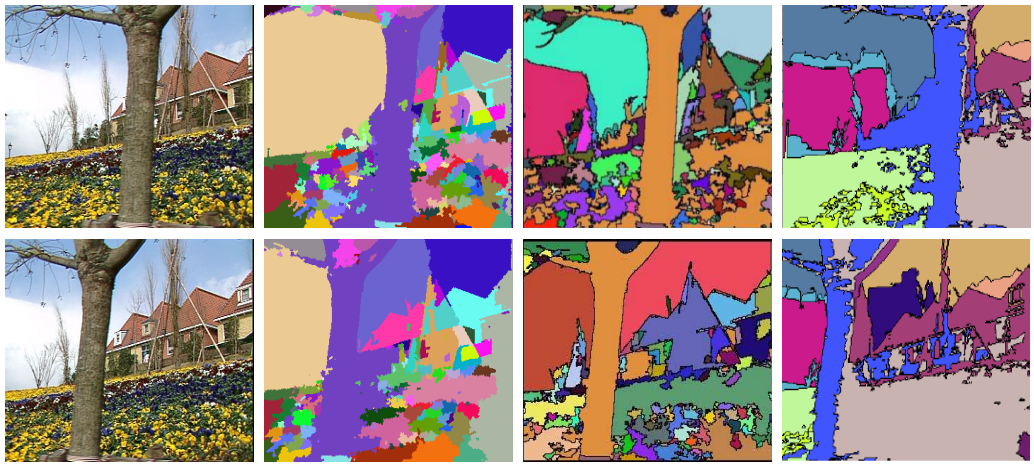


Figure 2.7: Video segmentation results for *waterski*, *goodfellas*, *ice skate* and *foroldmen* videos, shown here for qualitative evaluation based on color-coded correspondences.



(a) (b) (c) (d)

Figure 2.8: Comparison with [17] and [19] on the *garden* sequence. Column (a) shows two frames from the original sequence. Columns (b, c, d) show the results of the proposed methods, [17] and [19], respectively. Correspondences are shown using the same color. Performance can be evaluated by checking whether corresponding parts have the same color and different parts have different colors. The *garden* sequence is considered difficult for video segmentation/tracking for two reasons: (1) four major parts (sky, house, garden and tree) are at different depths, thereby causing different motions since the camera is moving, and (2) extremely textured regions within the garden produce a huge number of unstable regions which are hard to track. Result from [17] successfully tracks the front tree but fails to track almost all other regions. Result from [19] over-merges the regions, e.g., tree and house in the top frame, as well as the garden which becomes a single region tube despite the texture variation within. Our results preserve the local texture without over-smoothing, while correctly tracking each part. The shift of regions with the same color is highly correlated to the image motion.

CHAPTER 3

ACTIVITY RECOGNITION

3.1 Introduction

Human activity recognition is an exciting computer vision problem especially given today's ubiquitous mobile video recording devices. A reliable activity recognition system would have a huge impact on applications such as surveillance, video indexing, summarization and human computer interaction. In a nutshell, there are two recognition levels: in the first level, given a test video the system determines which activity occurs, a conventional recognition; in the next level the system further identifies *where* the activity occur, generally known as *localization*. Over the years lots of works are able to provide good results in the first recognition level [29, 30, 31]. In terms of video representation, most of the previous endeavor apply the "Bag-of-Words" (BoW) approach. That is, local patches (either 2D or 3D) are detected and represented by various feature extraction methods (e.g. gradient based, motion). Clusters among features are constructed, then a video is described by a distribution (e.g. histogram) of the feature clusters. Notwithstanding the demonstrated success on the first recognition level, the *orderless* nature for local patch based methods prevent further development. We believe a richer feature representation and more sophisticated model that entails relationships between features are crucial to the problem [32, 33].

A video can be seen as the aggregation of individual spatial-temporal volume (or tube) which is appearance and motion cohesive. An activity, a subset of the video, is comprised by a smaller set of tubes that have a consistent relationship. For now we assume the entire human body is always represented by either one or multiple tubes. Take a human waving activity for example to illustrate "consistent relationship". In the case of the entire human body

covered by one tube, the hand part of the tube would be consistently on the upper part of the tube. While if there are multiple tubes, the hand tube would be consistently on top of and adjacent to the torso tube. Certainly, a 3D spatial-temporal tube is a richer representation than a local patch, hence the problem can be reduced to two subproblems:
(1) how "reliable" tubes are obtained from a video, and
(2) how discriminant features are extracted from tubes.

Generic video segmentation has a strong boost in recent years [34, 35, 17, 9]. "Reliable" 3D segments can be obtained from the video segmentation. By reliable we mean in the context of activity recognition, the classification accuracy would not be jeopardized if small portions of the video get wrongly segmented as long as the majority "regions of interest" are well segmented.

In this dissertation, we propose a human activity recognition framework to address both conventional recognition and localization problems given spatial-temporal tubes as input. Inspired by the deformable parts model [36] in the context of object detection, we construct a tube-based Parts Activity Model (PAM) to address the spatial relationship *between* and *within* tubes. With the tube as a root level, we allow four rectangle parts down to the second level to achieve robust matching with imperfect input tubes in concern. Since an activity can be characterized as a 2D shape varies or deforms across time, the extracted feature of PAM consists of both *spatial shape* and *temporal dynamics* information. Instead of using feature distribution to describe a video like BoW, we treat each video as a set containing multiple tubes. To successfully classify a video and localize which tube(s) are the most discriminative, we formulate the problem as Multiple Instance Learning (MIL). By treating the most discriminative tube along with PAM parts location as latent variables, a max-margin discriminative learning approach Latent Support Vector Machine (LSVM) is applied. Hence, localization is achieved by the disclosure of the latent tube variable.

This chapter is organized as follows. After discussing several related works, we first introduce the Parts Activity Model (PAM) followed by a detail description of our tube-based feature extraction. The matching computation between the model and the instance will be shown as well. Second, the

mathematical formulation of the MIL framework and LSVM are described. Experimental results on two well-know datasets are presented in the Section 3.6.

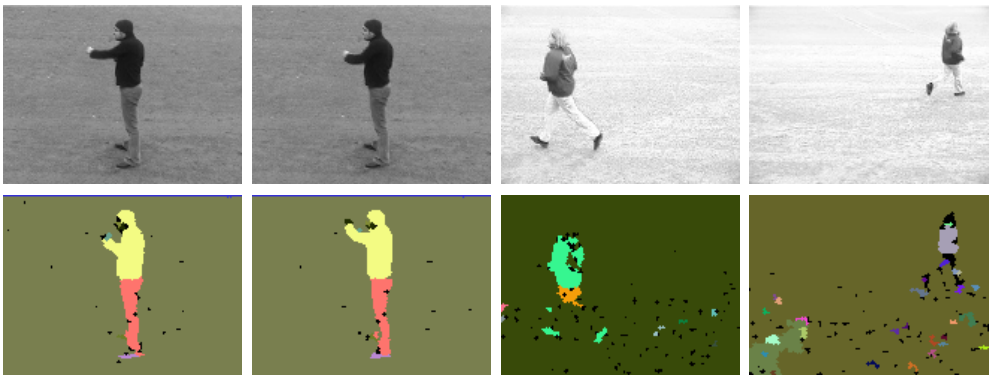
3.2 Related Work

The scope of activity recognition is huge, it covers video representation, feature extraction, model construction and learning approaches. To grasp all these different aspects, a through literature survey [37] is a good reference. Here we put our focus on using the spatial-temporal volume as a processing unit.

The core idea behind treating activity as a 3D spatial-temporal volume is that different activities in video generate distinct shapes (in terms of the 3D volume). Bobick and Davis [38] project the spatial-temporal volume down to motion-history images for simplicity, and Weinland et al. extended the work into motion-history volume [39]. These techniques work best when the action of interest is performed in a setting that enables reliable segmentation. In particular, for static scenes, techniques such as background subtraction can generate high-quality spatial-temporal volumes that are amenable to this analysis. Unfortunately, these conditions do not hold in typical real-world videos due to the presence of multiple moving objects and scene clutter. Similarly, the extensive research on generalizing shape matching [40, 41] requires reliable figure/ground separation. Ke et al. [42] presented one that closely relates to the line of our work. They used segmented spatio-temporal volumes to model human activities. Their system applies a hierarchical "MeanShift" to cluster similarly colored voxels, and it obtains several segmented volumes. However, in contrast to our approach, the models to be matched are derived from a single sample and they are manually constructed.

3.3 Parts Activity Model

As a tube-based activity model, in the ideal case each video contains a humanoid tube which represents a human silhouette varying across time. How-



(a) (b) (c) (d)

Figure 3.1: Over-segmented and over-smooth regions.

ever no matter how good the quality of the video segmentation algorithm, semantic human body movements could be divided into several tubes (*over-segment*), or there is no combination of tubes resulting in a semantic human body (*over-smooth*) as shown in Figure 3.1. In the first case, the divide can be caused by high appearance contrast within a human body (e.g. wearing a white shirt with black pants) or occlusion with other objects during the video. On the other hand, if there is drastic body motion which causes motion blur, or if no photometric or motion contrast exists, a 2D region boundary is not present in the image (frame) domain. In such cases, the human body part is merged with other tubes, hence no combination of tubes results in semantic human body.

Inspired by the deformable parts model developed in the work of object detection [36], we construct the Parts Activity Model (PAM). The model is a star structured. An activity is modeled in terms of tubes. For each segmented tube, there is an inherit 3D bounding box, which serves as the "root" after normalization, along with other rectangle "parts" where its positions are relative to the root (as shown in Figure 3.2). With a linear model assumption denoted as $\mathbf{w}$, which can be viewed as a weighted filter, the score of a tube $\mathbf{x}$ given a particular activity model $\mathbf{w}$ is $\mathbf{w}^T\Phi(\mathbf{x})$ where Φ extracts the feature vector of a tube (see details described in Section 3.3.1). The score $\mathbf{w}^T\Phi(\mathbf{x})$ comprises the score of the root filter, plus the score of part filters, and minus a deformation cost measuring the deviation of the part from its ideal position relative to the root filter. Both root and part filter scores are

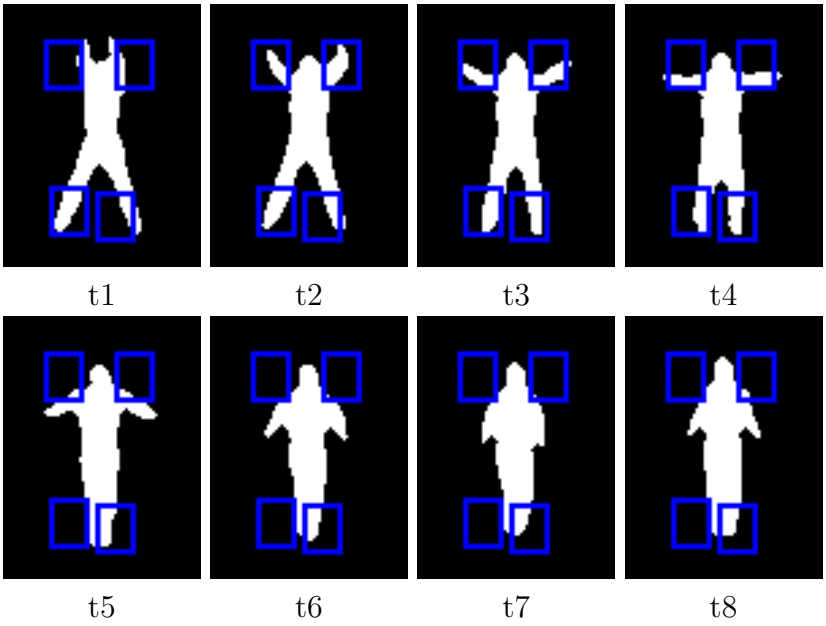


Figure 3.2: Parts Activity Model (PAM) of "jack" activity.

defined by a dot product between a filter and its corresponding target (i.e. root or part). Regarding the over-segmented scenario, while a test video consists of separated human body tubes (e.g. upper and lower body tubes), the matching score of either tube against the learned PAM is relatively high (compared to non-body-like tubes) since they are partially matched. Similar results also happen in the over-smooth scenario, in which the body-like tube will be emphasized (if one exists), however, the tube merged with the background would have a low score. Furthermore, activities may only differ in a small portion of the humanoid silhouette. The learned PAM can identify the discriminant part. Though the parts model is expected to handle the over-segmented and over-smooth tubes issues, lack of elaborated labeling on parts requires the model learning to have a set of *latent* or *hidden* variables $\mathfrak{p}$. We will address this problem in Section 3.4. The scoring function then is defined as:

$$f_{\mathbf{w}}(\mathbf{x}) = \max_{\mathfrak{p}} \mathbf{w}^T \Phi(\mathbf{x}, \mathfrak{p}) \tag{3.1}$$

The feature vector Φ is computed with the known part location $\mathfrak{p}$.

3.3.1 Robust Feature Extraction

A robust tube feature for activity recognition should have the following characteristics in addition to the discriminant power between different activities.
(1) *appearance or texture invariant*: the identity of activity should not be subject to the color of the clothing and the illumination condition of the environment. Therefore a tube is described as a 2D silhouette that varies across time.
(2) *scale, speed invariant*: video sequences can be taken at different focal lengths; even within a single video there may be zoom in/out effects, hence the feature should be scale invariant. We should also consider different speeds, which result from different frame rates.
(3) *length invariant*: the number of frames taken for the same activity should be invariant, i.e. taking 20 frames of a walking person should not change the feature of taking 200 frames of the same walking person. Since our segmented tube directly reflects the video content, tube length should not present a bias in the feature.

We propose to use *mean shape* plus the *frequency spectrum* as the tube features. Mean shape (shown in Figure 3.3) is used to describe the shape statistics both in root and parts, which is essentially the basic version of MHI (Motion History Image) [43]; the frequency spectrum is applied to describe the shape variation in terms of the root. Prior to computing the mean shape and frequency spectrum, the first task is to normalize each tube in terms of position, size and orientation. With a good normalization procedure, the mean shape of a given tube $\mathbf{x}$ can be easily computed as the average of all shapes (silhouettes) across time, denoted as

$$\phi_{ms}(\mathbf{x}) = \frac{1}{N_{\mathbf{x}}} \sum_{i=1}^{N_{\mathbf{x}}} s(x_i) \tag{3.2}$$

where $N_{\mathbf{x}}$ is the number of frames tube $\mathbf{x}$ survives, x_i is a particular slice of tube and $s(x_i)$ denotes the tube shape. The mean shape feature obtains the characteristics of appearance, speed and length invariant. Mean shape measures the statistics of shape along the time axis, but it alone cannot capture the dynamics of tube motion, that is, how (partial) shapes vary across time. The frequency spectrum provides the dynamics information.

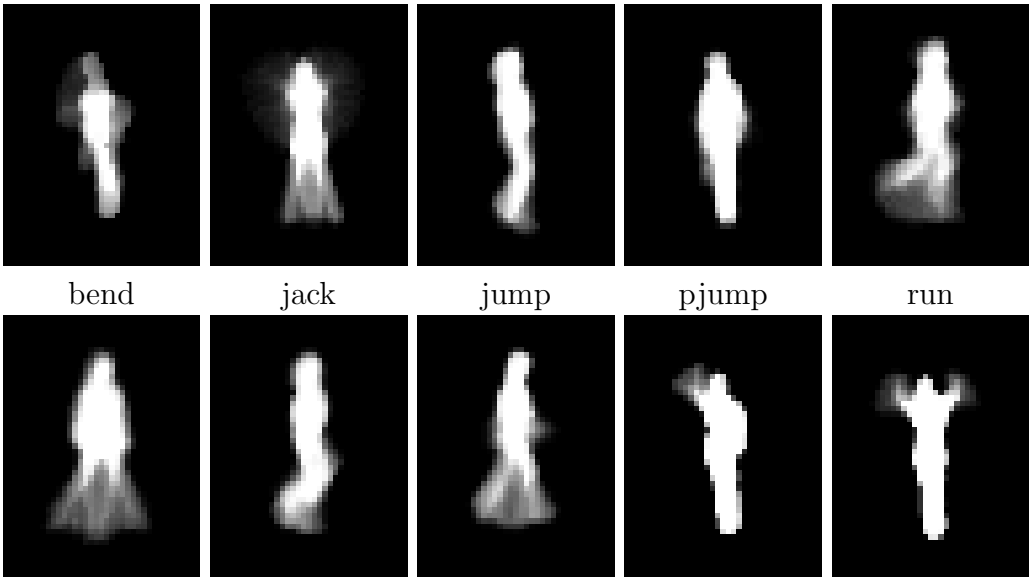


Figure 3.3: Visualization of the mean shape.

Frequency is a measurement of the root by its projection on the X and Y axes accordingly since the shape variation is expected to happen partially. We compute the number of pixels projected on the X (and Y) axes of each bin. A one-dimensional signal, i.e. number of pixels across time, of each bin is then constructed for Fourier frequency analysis. The frequency spectrum feature of a tube $\mathbf{x}$ consists of the Fourier frequency for all bins (on the X and Y axes).

$$\phi_f(\mathbf{x}) = [\mathrm{DFT}(\mathcal{B}_\mathrm{X}(\mathbf{x}))\ \mathrm{DFT}(\mathcal{B}_\mathrm{Y}(\mathbf{x}))] \tag{3.3}$$

DFT denotes Discrete Fourier Transform, $\mathcal{B}_\mathrm{X}$ denotes binning along the X axis and $\mathcal{B}_\mathrm{Y}$ denotes binning along the Y axis.

The above described mean shape and frequency spectrum are in the scope of intrinsic features, even the frequency spectrum is computed for each part of its essence to describe the individual part. A configuration feature is needed to provide joint model behavior. In our PAM model, we use parts deformation cost as a configuration feature. With the initial setting of anchor position (x_0, y_0) of each part for specific activity, let us define the *deformation* feature as:

$$\phi_d(dx, dy) = (dx, dy, dx^2, dy^2) \tag{3.4}$$

where $(dx_i, dy_i) = (x_i, y_i) - (x_0, y_0)$.

The final feature vector for a given tube $\mathbf{x}$ with parts information $\mathfrak{p} = \{p_1, p_2, ..., p_n\}$ is defined as:

$$\Phi(\mathbf{x}, \mathfrak{p}) = \Big(\phi_{ms}(\mathbf{x}), \{\phi_{ms}(\mathbf{x}, p_i)\}, \phi_f(\mathbf{x}), \{\phi_d(dx_i, dy_i)\}\Big) \tag{3.5}$$

Note that as shown in Equation (3.1), the scoring function is computed as a dot product between parameter $\mathbf{w}$ and the feature vector $\Phi(\mathbf{x}, \mathfrak{p})$. The first three sets of the parameters are essentially filters. While the last deformation parameters determine the cost of relative distances. For example, if it is $(0, 0, 1, 1)$, the deformation cost is the squared distance between its position and its anchor position.

3.4 Learning

Given our proposed parts activity model (PAM), the parameters expected to be learned consist of the mean shape feature, frequency feature, and parts displacement feature. To achieve that goal, one possibility is to learn from a dataset in which every video has detailed labels such as the "discriminant" tubes, part position and size of each tube, and on top of it is the conventional activity label. However there is no available activity dataset with such detailed human labels. Constructing such an elaborated dataset is expensive and time consuming. We decide to overcome the partially labeled dataset with sophisticated machine learning method. To achieve it, we introduce the idea of "hidden" or "latent" variables into the context. Simply put, latent variables represent model dependable variables without label information from the dataset. In our tube-based activity recognition scenario, latent variables include: (1) which tube(s) within a video has discriminant power, (2) location of parts for a given tube.

3.4.1 Multiple Instance Learning

To address the first part, discriminant tube within a video, of the latent variables, we apply a semi-supervised statistical method: Multiple Instance Learning (MIL) [44]. In a conventional statistical pattern recognition framework, it is usually assumed there is a training set with a labeled pair $(\mathbf{x}_i, y_i) \in \mathcal{R}^d \times \mathcal{Y}$ where $\mathbf{x}_i$ is a training instance and y_i is its label. The goal is to induce a classifier $f : \mathcal{R}^d \to \mathcal{Y}$. Here we assume it is a binary classifier in which $\mathcal{Y} \in \{-1, +1\}$ for simplicity. MIL generalizes the problem by making a significantly weaker assumption about the labeling information: the available label only exists in the *bag level* instead of the *instance level*. A bag B_I consists of a set of instances $\{\mathbf{x}_i : i \in I\}$ and associates with a label $Y_I \in \mathcal{Y}$. The definition of positive bag is that there exists at least one instance $\mathbf{x}_i \in B_I$ which is a positive instance, i.e. $y_i = 1$; otherwise it is a negative bag, i.e. $\forall \mathbf{x}_i \in B_I, y_i = -1$.

The configuration of MIL perfectly fits our tube-based activity recognition problem: a video sequence is a bag, and the segmented tubes within the video are the instances. In a binary case, a video with a positive activity label conveys there exists at least one tube that represents the characteristics of the activity; on the other hand, tubes within a negative video provide no strong candidates. There are different realizations of the above MIL configuration [45, 46, 47]. In general, they can be divided into two groups by whether they induce the instance label y_i. For a negative bag, the instance label is trivial. However, if the goal includes inducing an instance label of positive bags, then each y_i for the positive bags is treated as a variable to be learned. The advantage of such approaches is that both the bag and the instances within the bag can be classified. However, it will blow out the scope of the learned variables and result in slow convergence or a sub-optimal solution. Methods that do not induce an instance label can usually provide the confidence score of an instance that belonged to each label, thus the label can be inferred. Furthermore, it is likely that there are instances that do not belong to any specific label. Forcing those instances to be assigned to any label during the learning process would jeopardize the learned model. For example, background tubes among videos do not belong to a specific activity. It could be addressed by creating an artificial null label for the background tubes,

however the intra-class variance is expected to be huge. Therefore we chose to apply the bag level MIL approach in our activity recognition framework.

Maximum Bag Margin Formulation

Maximum margin approaches are widely used in classification works; one of the most popular ones is Support Vector Machine (SVM) [48]. The value of the maximum margin approach is that it learns a hyperplane which maximizes the distance between two "supported" hyperplanes of each class. In SVM, the instances that define the supported hyperplane are called "support vectors". The maximization of the margin provides good generalization.

Let $\mathbf{w}$ denote the hyperplane parameter to be learned on a conventional dataset with $(\mathbf{x}_i, y_i)$ pairs, where b is bias term, ξ is slack variable and C is the parameter which controls the tradeoff between the date term and regularization term. The classical SVM formulation is:

$$\begin{aligned} \min_{\mathbf{w},b} \ & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \\ \text{s.t. } & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \ \forall i \end{aligned} \tag{3.6}$$

In the bag level MIL configuration where the label is associated with bag, the maximum margin formulation becomes:

$$\begin{aligned} \min_{\mathbf{w},b} \ & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_I \xi_I \\ \text{s.t. } & Y_I \max_{i\in I}(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_I, \xi_I \geq 0, \ \forall I \end{aligned} \tag{3.7}$$

Under this formulation, only the instance that gives maximum response matters, and the others are irrelevant. The learned hyperplane $(\mathbf{w}, b)$ maximizes the margin between the "most positive" instance and the "least negative" instance. In other words, a bag is represented by a single "witness" instance, which is treated as a latent variable in the learning process. Along with other latent variables in our PAM model, the unified learning methodology Latent Support Vector Machine (LSVM) is described in Section 3.4.2. Our approach resembles the work in [45] by applying a MIL framework on top of SVM, however the optimization step is carefully crafted to cope with differ-

ent levels of latent variables in the activity recognition application (detailed in Section 3.4.2).

3.4.2 Latent Support Vector Machine

As we mentioned in the beginning of Section 3.4, latent variables in our tube-based PAM model comprise (1) the index of the representative tube within a video, and (2) the parts location for each segmented tube. The introduced maximum bag margin MIL with one latent variable for each bag (i.e. the index of the representative tube) can be viewed as a special case of LSVM. Before the actual LSVM formulation, we first rewrite the SVM equation for simplicity. Equation (3.6) can be written as:

$$\min_{\mathbf{w}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i), \ \forall i \tag{3.8}$$

This is equivalent with Equation (3.6) by assigning $\mathbf{x}^T \Leftarrow [\mathbf{x}^T \ 1]$, and $\mathbf{w}^T \Leftarrow [\mathbf{w}^T \ b]$.

Since the beginning of the section, $\mathbf{x}$ has been treated as a data instance to be classified. From now through the end of the section, we put our PAM model and tube-based activity recognition back into context: a video sequence is denoted as B, the tube within a video is $\mathbf{x}$, and the feature extracted from a tube $\mathbf{x}$ given parts information $\mathfrak{p}$ is $\Phi(\mathbf{x}, \mathfrak{p})$ as described in Section 3.3.1. Let $\Psi(B, z)$ denote the feature extraction function for video B given $z \in Z(B)$, which enumerates all combinations of the representative tube index and parts information. Given parameter $\mathbf{w}$, the scoring function $f_{\mathbf{w}}(I)$ of a video B_I is defined as:

$$\begin{aligned} f_{\mathbf{w}}(I) &= \max_{i \in I, \mathfrak{p}} \mathbf{w}^T \Phi(\mathbf{x}_i, \mathfrak{p}) \\ &= \max_{z \in Z(B_I)} \mathbf{w}^T \Psi(B_I, z) \end{aligned} \tag{3.9}$$

Hence the LSVM formulation is

$$\min_{\mathbf{w}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_I \max(0, 1 - Y_I f_{\mathbf{w}}(I)), \ \forall I \tag{3.10}$$

By comparing Equation (3.10) with Equation (3.7), we can easily see that the maximum bag margin MIL is a special case for LSVM if there is only one choice for z.

Optimization

As shown in Equation (3.10), the optimal parameter value $\mathbf{w}^*$ is achieve by minimizing the objective function $L_D(\mathbf{w})$ of training data D:

$$L_D(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_I \max(0, 1 - Y_I f_{\mathbf{w}}(I)),\ \forall I \tag{3.11}$$

If $L_D(\mathbf{w})$ is convex, then the optimal value can be achieved via the convex optimization approach. However, even if $f_{\mathbf{w}}(I)$ is linear, $L_D(\mathbf{w})$ is not convex, instead it is a *semi-convex* function: $L_D(\mathbf{w})$ is convex if $Y_I = -1$ and is concave if $Y_I = 1$. Recall that the max operation of convex functions is still convex. Given $f_{\mathbf{w}}(I)$ is linear for $\mathbf{w}$ which is convex, therefore when $Y_I = -1$ the hinge loss $\max(0, 1 + f_{\mathbf{w}}(I))$ is convex. However, $\max(0, 1 - f_{\mathbf{w}}(I))$ is concave in the case of a positive bag.

Let Z_p be a specific latent variable value for positive examples, meaning that once Z_p is set, we have information regarding the representative tube index and its part location for all positive bags. Then $L_D(\mathbf{w}, Z_p) = L_{D(Z_p)}(\mathbf{w})$, where $D(Z_p)$ is derived from dataset D by limiting the latent values for positive examples to Z_p. We can state:

$$L_D(\mathbf{w}) = \min_{Z_p} L_D(\mathbf{w}, Z_p) \tag{3.12}$$

Hence $L_D(\mathbf{w}) \leq L_D(\mathbf{w}, Z_p)$. By minimizing the upper bound $L_D(\mathbf{w}, Z_p)$, which is a convex function, the original semi-convex $L_D(\mathbf{w})$ is therefore minimized to local minima. Note that the latent variables come from two categories, one is the tube index of the video (the most positive tube since only positive examples are in the scope) and the other is the location of parts. Let Z_{pt} be specific tube index latent variables for positive examples; and let Z_{pl} be the specific parts location latent variables for positive examples. Equation

(3.12) can be extended:

$$\begin{aligned} L_D(\mathbf{w}) &= \min_{Z_p} L_D(\mathbf{w}, Z_p) \\ &\leq \min_{Z_{pl}} \min_{Z_{pt}} L_D(\mathbf{w}, Z_{pt}, Z_{pl}) \end{aligned} \tag{3.13}$$

The overall optimization procedure is *coordinate descent* by looping the following three steps:
(1): Optimize $L_D(\mathbf{w}, Z_{pt}, Z_{pl})$ over Z_{pt} by choosing the highest scoring latent tube index for each positive bag, $z_{I_t} = \arg\max_{z_{pt}} \mathbf{w}^T \Psi(B_I, z_{pt}, z_{pl})$, $\forall I, Y_I = 1$.
(2): Optimize $L_D(\mathbf{w}, Z_{I_t}, Z_{pl})$ over Z_{pl} by choosing the highest scoring latent parts location value for a given positive tube, $z_{I_l} = \arg\max_{z_{pl}} \mathbf{w}^T \Psi(B_I, z_{I_t}, z_{pl})$, $\forall I, Y_I = 1$.
(3): Optimize $L_D(\mathbf{w}, Z_{pt}, Z_{pl})$ over $\mathbf{w}$ by solving the convex optimization (here we use stochastic gradient method).

Though we are minimizing a looser upper bound $L_D(\mathbf{w}, Z_{pt}, Z_{pl})$ instead of $L_D(\mathbf{w}, Z_p)$, the local minima does not suffer much and the search space is significantly narrowed (from exponential to polynomial). The reason is that the tube index is far more important than the parts location. In other words, the parts location only matters when they are under the correct tube. The detailed optimization algorithm is described as Algorithm 4. The step size of the sub-gradient descent (line 17) is defined as:

$$h(\mathbf{w}, B_I, Y_I) = \begin{cases} 0 & \text{if } Y_i f_{\mathbf{w}}(I) \geq 1 \\ -Y_I \Psi(B_I, z_I(\mathbf{w})) & \text{if otherwise} \end{cases} \tag{3.14}$$

and α_t is the learning rate.

3.5 Implementation Details

The output tubes from video segmentation are used as input in the video recognition experiment. Since the discriminant tubes within a given video are treated as latent variables in our approach, filtering out unlikely tubes early in the process would contribute not only to accuracy but also ease

Algorithm 4 $\mathbf{w}^* = \text{optimizate } L_D(\mathbf{w}, Z_{pt}, Z_{pl})$

initialize $\mathbf{w}$, Z_{pl}
while *yet converge* **do**
    % step1: optimize over Z_{pt}
    $z_{I_t} = \arg\max_{z_{pt}} \mathbf{w}^T \Psi(B_I, z_{pt}, Z_{pl}), \forall I, Y_I = 1$
    % step2: optimize over Z_{pl}
    $z_{I_l} = \arg\max_{z_{pl}} \mathbf{w}^T \Psi(B_I, z_{I_t}, z_{pl}), \forall I, Y_I = 1$

    % step3: optimize over $\mathbf{w}$
    while *yet converge* **do**
        % optimize all latent variables for negative examples
        $z_{I_n} = \arg\max_z \mathbf{w}^T \Psi(B_I, z), \ \forall I, Y_I = -1$

        $z_I(\mathbf{w}) = [z_{I_t}, z_{I_l}, z_{I_n}]$
        $f_{\mathbf{w}}(I) = \mathbf{w}^T \Psi(B_I, z_I(\mathbf{w})), \forall I$

        % sub-gradient descent
        $\nabla L = \mathbf{w} + C \sum_I h(\mathbf{w}, B_I, Y_I)$
        $\mathbf{w} = \mathbf{w} - \alpha_t \nabla L$
    end while
end while
$\mathbf{w}^* = \mathbf{w}$

of computation in the optimization procedure. Therefore in practice, we insert a "relabeling" process between the video segmentation and recognition machinery. First we tried to discard tubes associated with fractional volume in the video. However, simply discarding them could jeopardize the shape feature of the tubes since it normally happens in the boundary on the object parts. To be more specific, given that our video segmentation takes the ramp-based over-segmented 2D regions as input, as mentioned in Section 2.2.1, object boundaries tend to be less stable, hence producing fragmented tubes. The relabeling process is shown in Algorithm 5.

3.6 Experimental Results

We conducted experiments on three well-known activity datasets: the Weizmann human action dataset [49], the KTH human motion datatset [50] and the UCF sports activity dataset [51] listed in order of difficulty. KTH is con-

Algorithm 5 relabeling

1: X : set of all tubes
2: X_f : set of fractional tubes
3: A : tube to label adjacency matrix
4: **for** $x_i \in X_f$ **do**
5: $\quad l^* = \arg\max_{X_j}(\text{volume}(X_j))$ where$A_{ij} = 1$
6: $\quad X_{l^*} = X_{l^*} \cup x_i$, % X_{l^*} is the set of tubes with label l^*
7: $\quad X_f = X_f \setminus x_i$
8: $\quad$ update A
9: **end for**

sidered more difficult than Weizmann due to its low resolution and variation camera movement, as well as its long duration and the appearance/texture of the subjects. UCF is even more difficult due to its various background textures and fast moving sports activity. For each dataset, both quantitative and qualitative results are provided. For quantitative results, we report per-video classification accuracy by using a leave-one-out setting. In the qualitative results, since our method processes on a tube base, we demonstrate localization of the most discriminant tube(s) within a video. The learned parts location is displayed to reveal which parts of the tubes are considered discriminant. Furthermore, visualization of the mean shape filter parameter is also shown.

Weizmann

The Weizmann dataset contains 93 videos from 9 subjects with 10 activities. Example video of each activity is shown at Appendix A. In Table 3.1, we compare our method with other representative methods. Since our method operates on tube-based video input after video segmentation, the accuracy is available in the per-video category. We achieve the state-of-the-art accuracy using the Weizmann dateset. The mean shape part of the learned parameter $\mathbf{w}$ is visualized in Figure 3.4. Since the multi-class classification is trained using a combination of binary classifiers, there are $\binom{10}{2}$ binary classifiers. The way to interpret each mean shape visualization is to look at the "relative" different parts. For example, 6 vs. 10 ("side" vs. "wave two hands"), the discriminant parts are the waving hands (for wave) and the horizontal leg movement (for side), therefore they are visually different than the other

Table 3.1: Comparison of classification accuracy with other previous works using the Weizmann dataset.

method	pre-frame	per-video	per-cube
Our method	N/A	**1.0000**	N/A
Wang & Mori 1[52]	0.9311	**1.0000**	N/A
Wang & Mori 2[53]	0.9029	0.9722	N/A
Jhuang et al.[54]	N/A	0.9880	N/A
Neibles & Fei-Fei[55]	0.5500	0.7280	N/A
Blank et al.[56]	N/A	N/A	0.9964

parts. So as for 5 vs. 8 ("run" vs. "walk"), since "run" has extended hands and legs motion compared to "walk", those parts are then highlighted with different values (lighter or darker than the other parts). Furthermore, the most discriminant tubes and their corresponding part locations of a selected subset of test videos are shown in Figure 3.5.

KTH

The KTH dataset contains six activities, each activity includes 25 persons performing the activity in four different configurations (e.g. clothes, background, etc.). Hence in total there are 600 videos. The classification accuracy is reported in Table 3.2. The reported accuracy belongs to the same level of the state-of-the-art accuracy from [31]. One of the major benefits of using segmented video tubes is not only to provide an activity label for a given video, but also to identify segments from the video content where the activity occurs. In Figure 3.6 the process from video to segmentation to representative tube is shown. The third row of both the boxing and wave videos are the tube representative power map, where bright color indicates higher representative power.

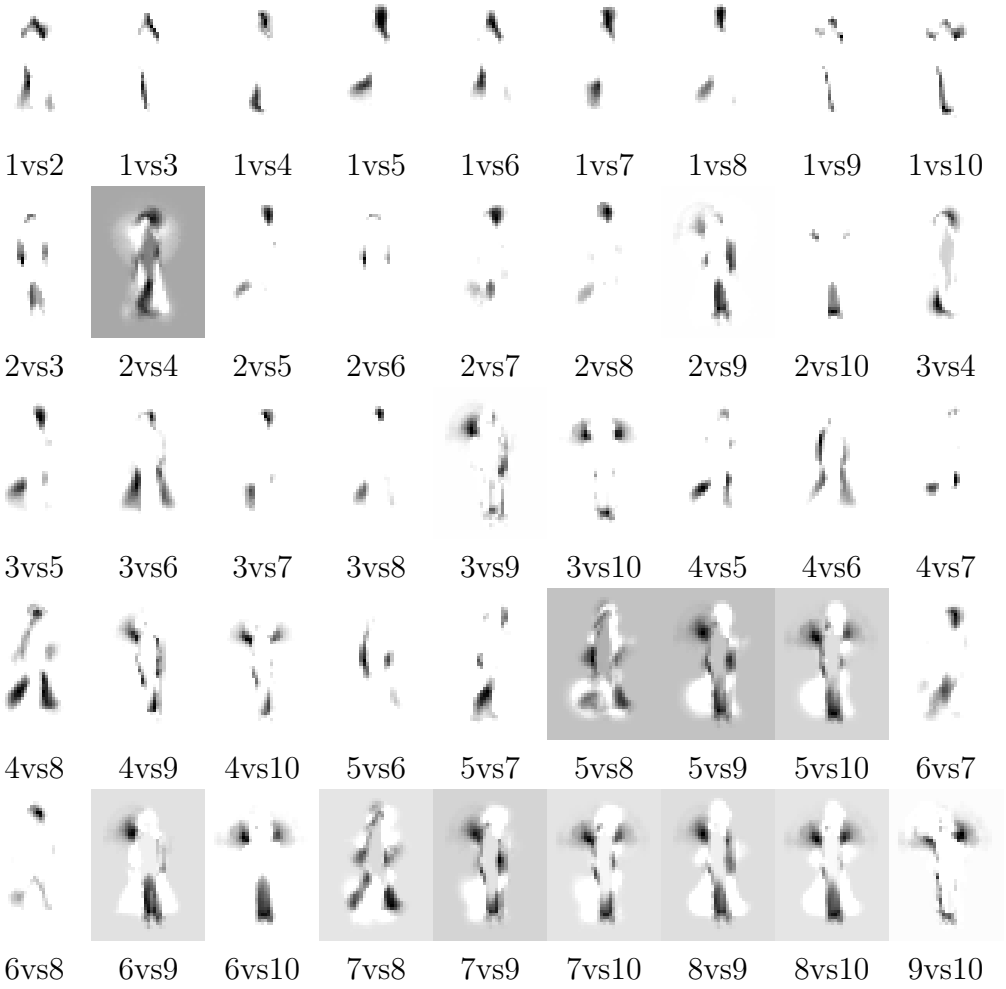


Figure 3.4: Learned mean shape parameter of binary classifiers of the Weizmann dataset. 1:bend, 2:jack: 3:jump, 4:pjump, 5:run, 6:side, 7:skip, 8:walk, 9:wave1, 10:wave2.

Table 3.2: Comparison of classification accuracy with other previous works on the KTH dataset.

method	**accuracy**
Our method	**0.9510**
Wang & Mori 1[53]	0.9251
Wang & Mori 2[52]	0.8760
Jhuang et at.[54]	0.9170
Neibles & Fei-Fei[55]	0.8150
Liu & Shah[31]	**0.9416**
Dollár et at.[57]	0.8117
Schuldt et al.[58]	0.7172

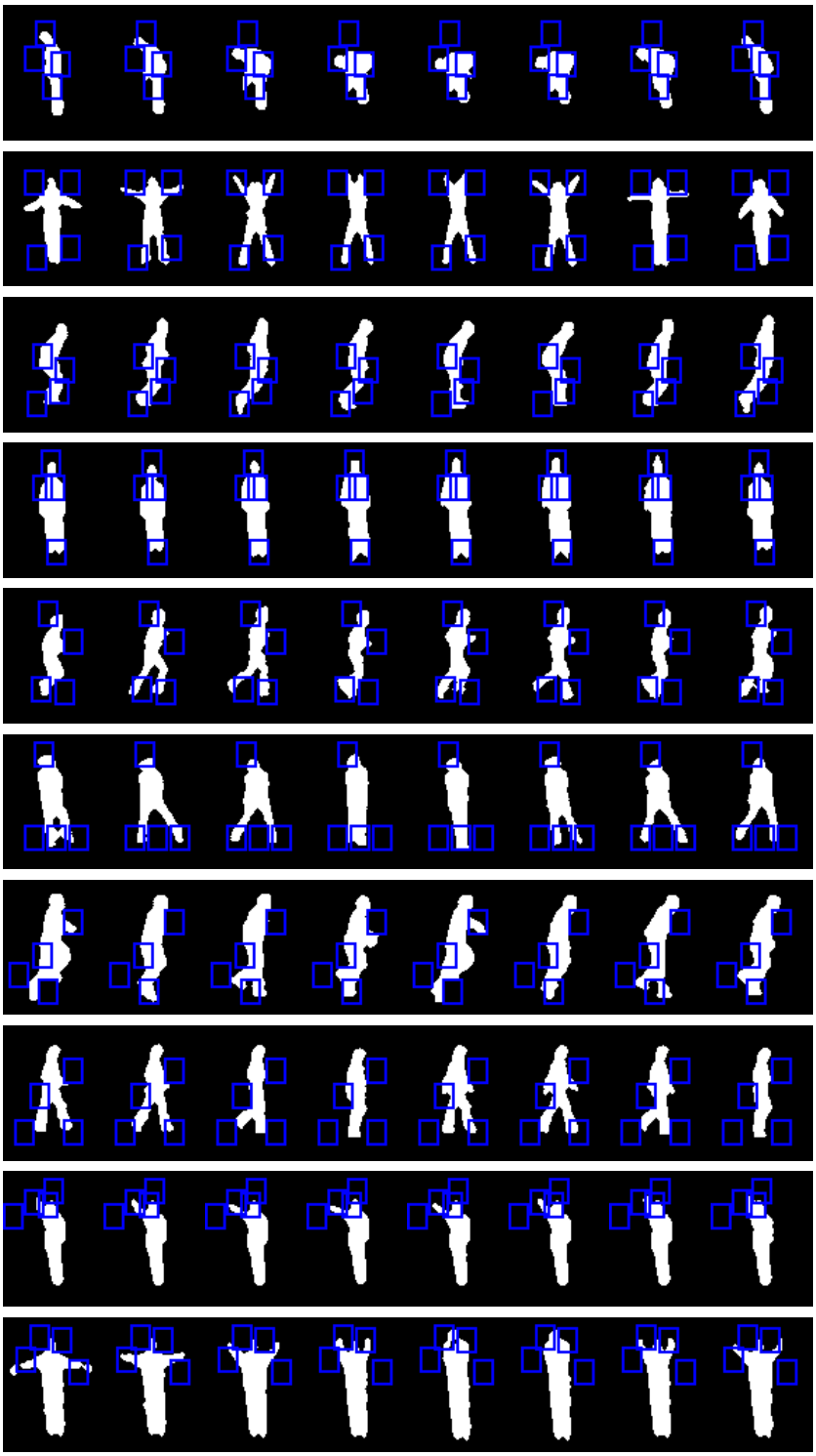


Figure 3.5: Discriminant tubes of a selected subset of test videos across activities on the Weizmann dataset. The parts locations which induce the best score are displayed. The activity videos from top to bottom are called bend, jack, jump, pjump, run, side, skip, walk, wave1 and wave2.

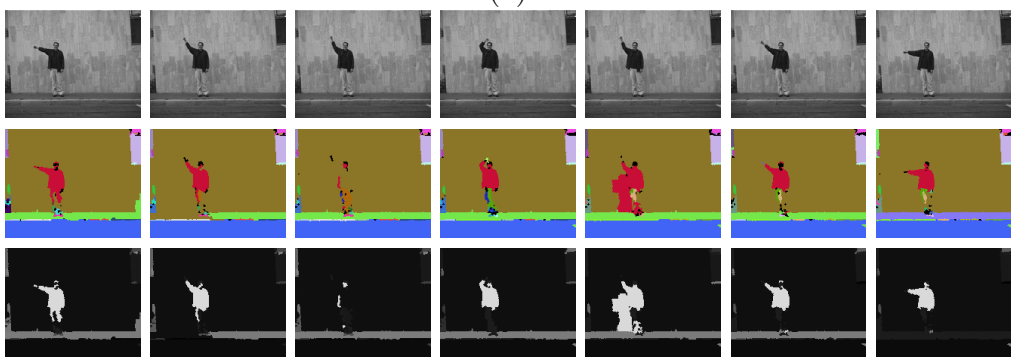
Table 3.3: Confusion matrix of the UCF dataset. There are more than 200 videos across nine activities. The experiment is conducted in leave-one-out fashion.

	diving	golf swinging	kicking	lifting	riding	running	skating	swinging	walking
diving	0.95	0	0	0	0.05	0	0	0	0
golf swinging	0	0.65	0.08	0	0	0	0	0	0.27
kicking	0.04	0.10	0.72	0	0	0	0	0	0.14
lifting	0	0	0	0.71	0	0	0	0	0.29
riding	0.20	0	0.05	0	0.75	0	0	0	0
running	0	0	0	0	0	0.71	0.11	0.05	0.03
skating	0	0	0	0	0	0.07	0.93	0	0
swinging	0	0.02	0	0	0	0	0.03	0.95	0
walking	0	0	0	0	0.05	0.07	0	0	0.88

UCF

Lastly, we conducted experiments on the UCF sports dataset, which consists of more than 200 sports video across nine different sporting activities collected by and shown in [51]. It includes diving, golf swinging, kicking, lifting, riding, running, skating, swinging and walking. The UCF dataset brought the most challenge to our video segmentation based approach due to faster athletic movement and a textured background. The detailed confusion matrix we obtained for this dataset is depicted in Table 3.3. The overall mean accuracy we obtain is 80.55%, compared to 79.2% reported in [59], 85.6% in [60], 87.27% in [32] and 69.2% in the original paper [51]. The main reason for the noticeable accuracy difference compared to [60] and [32] is the qualify of video segmentation outputs. Since our approach is tightly coupled with video segmentation outputs, the variational background in the dataset creates noisy tubes, an example is shown at Figure 3.7. Notwithstanding unavoidable noisy tubes, the major contribution of coupled segmentation + recognition framework is the ability to locate regions corresponding to the activity. As shown in Figure 3.8, we successfully located the hands and legs regions of the diving activity.

(a)



(b)

Figure 3.6: Representative tubes. Two example video sequences, boxing (KTH) and wave (Weizmann) illustrate the inferred representative tubes. In both video examples, the first row is the input sequence; the second row is the color-coded video segmentation result, a tube is associated with a specific color; the third row is the weight map which corresponds to the representative power of the tubes. In (a) boxing, a human is segmented into two tubes (upper and lower body). Through our Multiple Instance Learning mechanism, the upper body tube is identified as the most representative tube among all tubes. As for the wave example in (b), we demonstrate an imperfect video segmentation result: in the third column the human body is missing, in the fifth column a part of the background is merged into the upper body; the most representative tube is still correctly identified. It suggests the proposed PAM model and learning mechanism could deal with noisy video segmentation results, which is common for more complicated video/activity.

Figure 3.7: Example of a noisy tube from video segmentation output. The left figure is the original frame. The right figure displays regions (colored in red) from a noisy tube: the color of the subject's upper body is very close to the background, hence they belong to the same tube.

Figure 3.8: Example of an output discriminative tube (in a single frame). The left figure is the original frame. Red-colored regions of the right figure are the output discriminative tubes in a specific frame.

CHAPTER 4

CONCLUSION

We addressed the fundamental problem of computer vision: segmentation and recognition, in the space-time domain. With the knowledge that generic image segmentation introduces unstable regions due to illumination, compression, etc., we utilized temporal information to achieve consistent 3D video segmentation. By exploiting non-local structure in both spatial and temporal space, the instabilities of the segmented regions were alleviated. A segmentation tree was built within every frame, the label consistency was enforced within each subtree (i.e. spatial clique). By roughly tracking 2D regions across each frame, temporal clique was built in which label consistency was enforced as well. The high-order (more than binary) Conditional Random Field (CRF) is designed and solved efficiently. Experimental results demonstrate high-quality segmentation quantitatively and qualitatively.

Taking segmented 3D regions, called tubes, as input, we developed an activity recognition framework not only to determine which activity existed in a video but also to locate where it happens. A robust tube feature was extracted with photometric and shape dynamics information. Activity was described as a Parts Activity Model (PAM) with a root template and four parts template under the root. Given the nature of the activity recognition problem that only some parts on the video were used to determine the activity label, we used Multiple Instance Learning (MIL) to formulate the problem. Latent variables included tube index and the parts location under the root template. Experiments were conducted on three well-known dataset and a state-of-the-art result was achieved.

APPENDIX A

DATASET PREVIEW

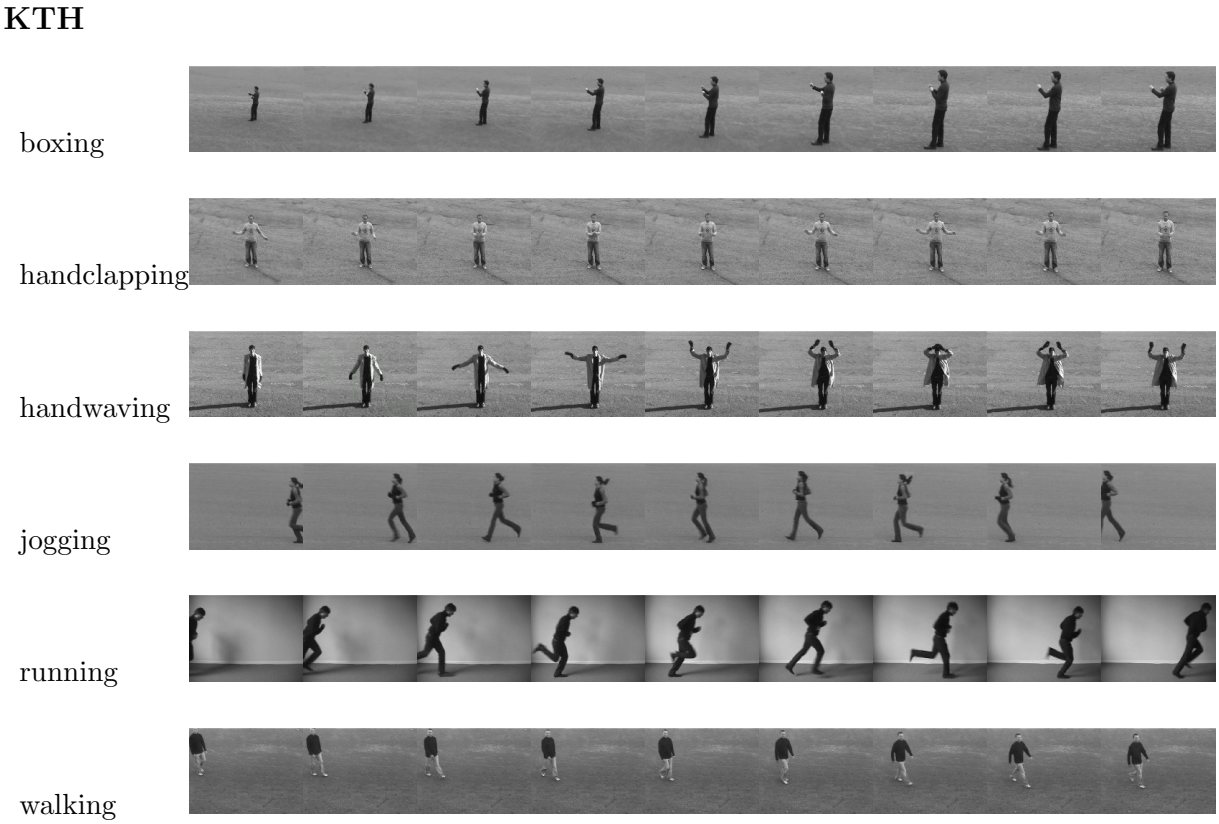


Figure A.1: Video examples for each activity of the KTH dataset.

Weizmann

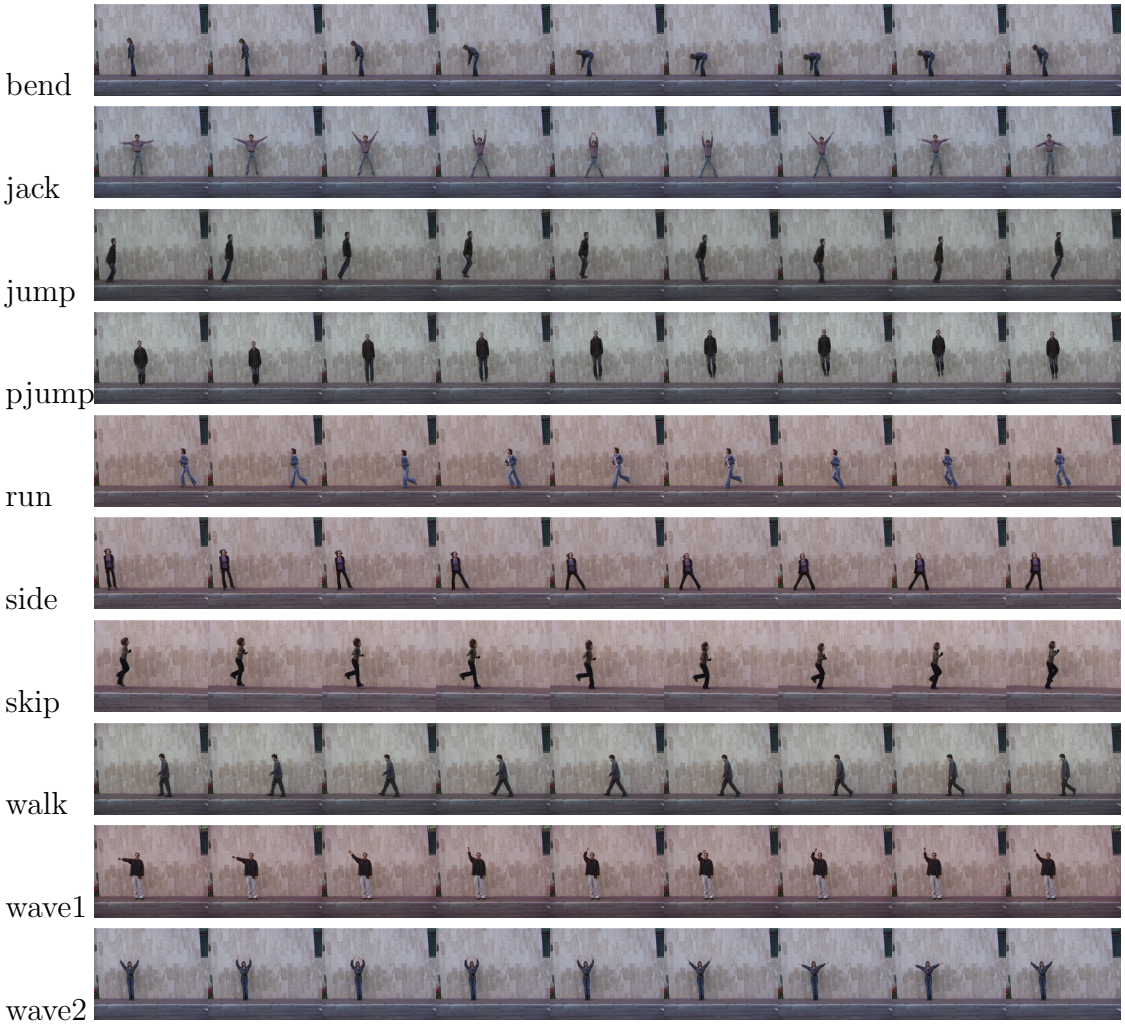


Figure A.2: Examples of the Weizmann human action dataset.

REFERENCES

[1] A. Torralba, K. Murphy, and W. Freeman, "Sharing features: Efficient boosting procedures for multiclass object detection," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2004.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," in *IJCV, International Journal on Computer Vision*, 2004.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2005.

[4] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering object categories in image collections," in *ICCV, Proceedings of International Conference on Computer Vision*, 2005.

[5] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2005.

[6] N. Ahuja, "A transform for multiscale image segmentation by integrated edge and region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 12, pp. 1211–1235, December 1996.

[7] E. Akbas and N. Ahuja, "From ramp discontinuities to segmentation tree," in *ACCV, Proceedings of Asian Conference on Computer Vision*, 2009.

[8] S. Todorovic and N. Ahuja, "Unsupervised category modeling, recognition, and segmentation in images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 12, pp. 2158–2174, December 2008.

[9] S. Paris, "Edge-preserving smoothing and mean-shift segmentation of video streams," in *ECCV, Proceedings of European Conference on Computer Vision*, 2008.

[10] J. Kim and J. Woods, "Spatiotemporal adaptive 3-d Kalman filter for video," *IEEE Trans. Image Processing*, vol. 6, no. 3, pp. 414–424, 1997.

[11] J. Y. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 625–638, 1994.

[12] J. Wang, B. Thiesson, Y. Xu, and M. Cohen, "Image and video segmentation by anisotropic kernel mean shift," in *ECCV, Proceedings of European Conference on Computer Vision*, 2004.

[13] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nystrom method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26(2), pp. 214–225, 2004.

[14] H. Greenspan, J. Goldberger, and A. Mayer, "A probabilistic framework for spatio-temporal video representation," in *ECCV, Proceedings of European Conference on Computer Vision*, 2002.

[15] S. Khan and M. Shah, "Object based segmentation of video using color, motion and spatial information," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2001.

[16] C. L. Zitnick, N. Jojic, and S. B. Kang, "Consistent segmentation for optical flow estimation," in *ICCV, Proceedings of International Conference on Computer Vision*, 2005.

[17] W. Brendel and S. Todorovic, "Video object segmentation by tracking regions," in *ICCV, Proceedings of International Conference on Computer Vision*, 2009.

[18] V. Hedau and N. Ahuja, "Matching images under unstable segmentations," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2008.

[19] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2010.

[20] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller, "Multiple hypothesis video segmentation from superpixel flows," in *ECCV, Proceedings of European Conference on Computer Vision*, 2010.

[21] P. Kohli, L. Ladicky, and P. Torr, "Robust higher order potentials for enforcing label consistency," *IJCV, International Journal on Computer Vision*, vol. 82, no. 3, pp. 320–324, 2009.

[22] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, November 2001.

[23] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML, Proceedings of International Conference on Machine Learning*, 2001.

[24] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, March 2011.

[25] L. Gorelick, M. Blank, E.Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 2247–2253, 2007.

[26] H. Zhang, J. Fritts, and S. Goldman, "Image segmentation: A survey of unsupervised methods," in *CVIU, Computer Vision and Image Understanding*, 2009.

[27] C. Erdem, B. Sankur, and A. M. Tekalp, "Performance measures for video object segmentation and tracking," *IEEE Trans. Image Processing*, vol. 13, pp. 947–51, 2004.

[28] H. Zhang, J. Fritts, and S. Goldman, "An entropy-based objective evaluation method for image segmentation," in *SPIE*, 2003.

[29] J. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2007.

[30] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2008.

[31] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2009.

[32] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood feature for human action recognition," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2010.

[33] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 288–303, 2010.

[34] H. Cheng and N. Ahuja, "Exploiting nonlocal spatiotemporal structure for video segmentation," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2012.

[35] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2010.

[36] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.

[37] J. Aggarwal and M. Ryoo, "Human activity analysis," *ACM Computing Surveys*, vol. 43, no. 3, 2011.

[38] J. Bobick and J. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 3, pp. 257–267, 2001.

[39] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 249–257, 2006.

[40] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, "Shape representation and classification using the Poisson equation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1991–2005, 2006.

[41] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 286–299, 2007.

[42] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," in *ICCV, Proceedings of International Conference on Computer Vision*, 2007.

[43] M. A. Ahad, J. K. Tan, H. Kim, and S. Ishikawa, "Motion history image: Its variants and applications," *Machine Vision and Applications*, vol. 23, no. 2, 2012.

[44] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez, "Solving the multiple instance problem with axis-parallel rectangles," in *Artificial Intelligence*, 1997.

[45] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *NIPS*, 2002.

[46] T. Gartner, P. A. Flach, A. Kowalczyk, and A. J. Smola, "Multi-instance kernels," in *ICML, Proceedings of International Conference on Machine Learning*, 2002.

[47] J. Ramon and L. D. Raedt, "Multi instance neural networks," in *Workshop of Attribute-Value and Relational Learning, ICML*, 2000.

[48] B. Scholkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. Cambridge, MA: MIT Press, 2002.

[49] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, December 2007.

[50] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *ICPR, Proceedings of International Conference on Pattern Recognition*, 2004.

[51] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2008.

[52] Y. Wang and G. Mori, "Max-margin hidden conditional random fields for human action recognition," in *CVPR, Proceedings of Computer Vision and Pattern Recognition*, 2009.

[53] Y. Wang and G. Mori, "Learning a discriminative hidden part model for human action recognition," in *NIPS*, 2008.

[54] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *ICCV, Proceedings of International Conference on Computer Vision*, 2007.

[55] J. Neibles, C. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *ECCV*, 2010.

[56] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *ICCV, Proceedings of International Conference on Computer Vision*, 2005.

[57] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[58] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *ICPR, Proceedings of International Conference on Pattern Recognition*, 2004.

[59] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," in *ICCV, Proceedings of International Conference on Computer Vision*, 2009.

[60] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009.